Advanced Gtk+ Sequencer Audio Reference Manual

Mr. Joël Krähemann

Copyright (C)

COLLABORATORS

	TITLE : Advanced Gtk+ Sequencer Audio Reference Manual	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Mr. Joël Krähemann	April 2, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

I Plugin - API of plugins 1

1 Base plugin types 3

1.1 AgsBasePlugin . . . 3
1.2 ags_plugin_stock . . . 24
1.3 AgsPluginPort . . . 25

2 LADSPA plugins 39

2.1 AgsLadspaConversion . . . 39
2.2 AgsLadspaManager . . . 44
2.3 AgsLadspaPlugin . . . 50

3 DSSI plugins 53

3.1 AgsDssiPlugin . . . 53
3.2 AgsDssiManager . . . 57

4 LV2 plugins 63

4.1 AgsLv2Conversion . . . 63
4.2 AgsLv2EventManager . . . 67
4.3 AgsLv2LogManager . . . 70
4.4 AgsLv2Manager . . . 73
4.5 AgsLv2OptionManager . . . 81
4.6 AgsLv2TurtleParser . . . 89
4.7 AgsLv2TurtleScanner . . . 93
4.8 AgsLv2Plugin . . . 97
4.9 AgsLv2PresetManager . . . 111
4.10 AgsLv2Preset . . . 113
4.11 AgsLv2UriMapManager . . . 120
4.12 AgsLv2UridManager . . . 124
4.13 AgsLv2Worker . . . 128
4.14 AgsLv2WorkerManager . . . 135
4.15 AgsLv2uiManager . . . 138
4.16 AgsLv2uiPlugin . . . 143

II The audio layer 149

5 Utility functions 151

5.1 AgsSoundEnums . . . 151
5.2 AgsSynthEnums . . . 157
5.3 AgsSoundcardUtil . . . 158
5.4 AgsSequencerUtil . . . 160
5.5 AgsCharBufferUtil . . . 161
5.6 AgsChorusUtil . . . 172
5.7 AgsLinearInterpolateUtil . . . 190
5.8 AgsFourierTransformUtil . . . 201
5.9 AgsFastPitchUtil . . . 211
5.10 AgsHQPitchUtil . . . 221
5.11 AgsTimeStretchUtil . . . 232
5.12 AgsPhaseShiftUtil . . . 245
5.13 AgsFrequencyAliaseUtil . . . 256
5.14 AgsAudioBufferUtil . . . 267
5.15 AgsVolumeUtil . . . 322
5.16 AgsEnvelopeUtil . . . 331
5.17 AgsPeakUtil . . . 341
5.18 AgsNoiseUtil . . . 352
5.19 AgsResampleUtil . . . 365
5.20 AgsSynthUtil . . . 375
5.21 AgsFMSynthUtil . . . 401
5.22 AgsSF2SynthUtil . . . 427
5.23 AgsSFZSynthUtil . . . 450
5.24 AgsLFOSynthUtil . . . 466
5.25 AgsSynthGenerator . . . 490
5.26 AgsSF2SynthGenerator . . . 513
5.27 AgsSFZSynthGenerator . . . 535

6 Utility functions from Fluidsynth 551

6.1 AgsFluidUtil . . . 551
6.2 AgsFluidInterpolateNoneUtil . . . 554
6.3 AgsFluidInterpolateLinearUtil . . . 567
6.4 AgsFluidInterpolate4thOrderUtil . . . 580
6.5 AgsFluidInterpolate7thOrderUtil . . . 593
6.6 AgsFluidIIRFilterUtil . . . 606

7 Frequency utils **621**
7.1 AgsFrequencyMapManager . . . 621
7.2 AgsFrequencyMap . . . 627

8 Audio sinks and sources **636**
8.1 AgsFifoout . . . 636
8.2 AgsAlsaDevin . . . 645
8.3 AgsAlsaDevout . . . 655
8.4 AgsAlsaMidiin . . . 664
8.5 AgsOssDevin . . . 673
8.6 AgsOssDevout . . . 683
8.7 AgsOssMidiin . . . 693
8.8 AgsWasapiDevout . . . 701
8.9 AgsWasapiDevin . . . 710
8.10 AgsJackServer . . . 720
8.11 AgsJackClient . . . 727
8.12 AgsJackPort . . . 735
8.13 AgsJackDevin . . . 741
8.14 AgsJackDevout . . . 751
8.15 AgsJackMidiin . . . 761
8.16 AgsPulseServer . . . 769
8.17 AgsPulseClient . . . 776
8.18 AgsPulsePort . . . 785
8.19 AgsPulseDevin . . . 792
8.20 AgsPulseDevout . . . 802
8.21 AgsGstreamerServer . . . 813
8.22 AgsGstreamerClient . . . 820
8.23 AgsGstreamerPort . . . 828
8.24 AgsGstreamerDevin . . . 836
8.25 AgsGstreamerDevout . . . 846
8.26 AgsCoreAudioServer . . . 856
8.27 AgsCoreAudioClient . . . 863
8.28 AgsCoreAudioPort . . . 871
8.29 AgsCoreAudioDevin . . . 878
8.30 AgsCoreAudioDevout . . . 888
8.31 AgsCoreAudioMidiin . . . 897
8.32 AgsAudioUnitServer . . . 905
8.33 AgsAudioUnitClient . . . 912
8.34 AgsAudioUnitPort . . . 920
8.35 AgsAudioUnitDevin . . . 927
8.36 AgsAudioUnitDevout . . . 937

9 The audio tree **948**

9.1 AgsSoundProvider . . . 948
9.2 AgsAudioApplicationContext . . . 955
9.3 AgsAudio . . . 958
9.4 AgsChannel . . . 1041
9.5 AgsOutput . . . 1096
9.6 AgsInput . . . 1099
9.7 AgsRecycling . . . 1104
9.8 AgsAudioSignal . . . 1122
9.9 AgsPort . . . 1150

10 Storage objects **1163**

10.1 AgsNotation . . . 1163
10.2 AgsNote . . . 1184
10.3 AgsAutomation . . . 1208
10.4 AgsAcceleration . . . 1236
10.5 AgsWave . . . 1244
10.6 AgsBuffer . . . 1264
10.7 AgsMidi . . . 1273
10.8 AgsTrack . . . 1284
10.9 AgsPattern . . . 1292
10.10 AgsPreset . . . 1303

11 Fundamental recall types **1318**

11.1 AgsFxFactory . . . 1318
11.2 AgsPlaybackDomain . . . 1321
11.3 AgsPlayback . . . 1328
11.4 AgsRecallID . . . 1335
11.5 AgsRecyclingContext . . . 1342
11.6 AgsRecallContainer . . . 1351
11.7 AgsRecallDependency . . . 1360
11.8 AgsRecall . . . 1364
11.9 AgsRecallAudio . . . 1416
11.10 AgsRecallAudioRun . . . 1420
11.11 AgsRecallChannel . . . 1424
11.12 AgsRecallChannelRun . . . 1429
11.13 AgsRecallRecycling . . . 1436
11.14 AgsRecallAudioSignal . . . 1440
11.15 AgsGenericRecallChannelRun . . . 1443
11.16 AgsGenericRecallRecycling . . . 1446

12 Audio thread - multi-threaded audio 1449

12.1 AgsAudioLoop . . . 1449
12.2 AgsAudioThread . . . 1456
12.3 AgsChannelThread . . . 1465
12.4 AgsExportThread . . . 1474
12.5 AgsSequencerThread . . . 1478
12.6 AgsSoundcardThread . . . 1481
12.7 AgsSF2Loader . . . 1484
12.8 AgsSF2MidiLocaleLoader . . . 1493
12.9 AgsSFZLoader . . . 1504
12.10 AgsSFZInstrumentLoader . . . 1511
12.11 AgsWaveLoader . . . 1520

13 Audio MIDI - MIDI support 1527

13.1 AgsMidiBufferUtil . . . 1527
13.2 AgsMidiFile . . . 1552
13.3 AgsMidiUtil . . . 1566
13.4 AgsMidiBuilder . . . 1578
13.5 AgsMidiParser . . . 1603

14 Audio OSC - OSC support 1634

14.1 AgsOscBufferUtil . . . 1634
14.2 AgsOscUtil . . . 1645
14.3 AgsOscBuilder . . . 1652
14.4 AgsOscParser . . . 1664

15 Audio OSC - OSC client 1677

15.1 AgsOscClient . . . 1677

16 Audio OSC - OSC server 1687

16.1 AgsOscServer . . . 1687
16.2 AgsOscConnection . . . 1699
16.3 AgsOscMessage . . . 1706
16.4 AgsOscResponse . . . 1710
16.5 AgsOscController . . . 1716
16.6 AgsOscActionController . . . 1719
16.7 AgsOscConfigController . . . 1722
16.8 AgsOscExportController . . . 1725
16.9 AgsOscFrontController . . . 1729
16.10 AgsOscInfoController . . . 1735

16.11 AgsOscMeterController . . . 1738
16.12 AgsOscNodeController . . . 1747
16.13 AgsOscPluginController . . . 1750
16.14 AgsOscRenewController . . . 1753
16.15 AgsOscStatusController . . . 1756

17 Audio OSC XMLRPC - OSC over XMLRPC server 1760

17.1 AgsOscXmlrpcServer . . . 1760
17.2 AgsOscWebsocketConnection . . . 1763
17.3 AgsOscXmlrpcMessage . . . 1767
17.4 AgsOscXmlrpcController . . . 1770

18 Audio files - Reading/Writing audio data from disk 1777

18.1 AgsAudioContainerManager . . . 1777
18.2 AgsAudioContainer . . . 1781
18.3 AgsAudioFileManager . . . 1793
18.4 AgsAudioFile . . . 1797
18.5 AgsAudioFileLink . . . 1815
18.6 AgsGstreamerFile . . . 1818
18.7 AgsIpatch . . . 1827
18.8 AgsIpatchGigReader . . . 1832
18.9 AgsIpatchSF2Reader . . . 1838
18.10 AgsIpatchDLS2Reader . . . 1846
18.11 AgsIpatchSample . . . 1853
18.12 AgsSndfile . . . 1857
18.13 AgsSFZFile . . . 1862
18.14 AgsSFZGroup . . . 1871
18.15 AgsSFZRegion . . . 1877
18.16 AgsSFZSample . . . 1884
18.17 AgsSoundContainer . . . 1894
18.18 AgsSoundResource . . . 1902

19 Fx Engine - playback and recording 1911

19.1 AgsFxPlaybackAudio . . . 1911
19.2 AgsFxPlaybackAudioProcessor . . . 1921
19.3 AgsFxPlaybackAudioSignal . . . 1925
19.4 AgsFxPlaybackChannel . . . 1928
19.5 AgsFxPlaybackChannelProcessor . . . 1930
19.6 AgsFxPlaybackRecycling . . . 1932
19.7 AgsFxBufferAudio . . . 1935

19.8 AgsFxBufferAudioProcessor . . . 1937
19.9 AgsFxBufferAudioSignal . . . 1939
19.10 AgsFxBufferChannel . . . 1942
19.11 AgsFxBufferChannelProcessor . . . 1947
19.12 AgsFxBufferRecycling . . . 1949

20 Fx Engine - pattern and notation 1952

20.1 AgsFxPatternAudio . . . 1952
20.2 AgsFxPatternAudioProcessor . . . 1957
20.3 AgsFxPatternAudioSignal . . . 1960
20.4 AgsFxPatternChannel . . . 1963
20.5 AgsFxPatternChannelProcessor . . . 1965
20.6 AgsFxPatternRecycling . . . 1967
20.7 AgsFxNotationAudio . . . 1970
20.8 AgsFxNotationAudioProcessor . . . 1977
20.9 AgsFxNotationAudioSignal . . . 1981
20.10 AgsFxNotationChannel . . . 1985
20.11 AgsFxNotationChannelProcessor . . . 1987
20.12 AgsFxNotationRecycling . . . 1989
20.13 AgsFxDssiAudio . . . 1992
20.14 AgsFxDssiAudioProcessor . . . 1999
20.15 AgsFxDssiAudioSignal . . . 2002
20.16 AgsFxDssiChannel . . . 2004
20.17 AgsFxDssiChannelProcessor . . . 2007
20.18 AgsFxDssiRecycling . . . 2009
20.19 AgsFxLv2Audio . . . 2011
20.20 AgsFxLv2AudioProcessor . . . 2019
20.21 AgsFxLv2AudioSignal . . . 2021
20.22 AgsFxLv2Channel . . . 2024
20.23 AgsFxLv2ChannelProcessor . . . 2027
20.24 AgsFxLv2Recycling . . . 2030
20.25 AgsFxSynthAudio . . . 2032
20.26 AgsFxSynthAudioProcessor . . . 2048
20.27 AgsFxSynthAudioSignal . . . 2050
20.28 AgsFxSynthChannel . . . 2053
20.29 AgsFxSynthChannelProcessor . . . 2055
20.30 AgsFxSynthRecycling . . . 2057
20.31 AgsFxFMSynthAudio . . . 2060
20.32 AgsFxFMSynthAudioProcessor . . . 2075

20.33 AgsFxFMSynthAudioSignal . . . 2077
20.34 AgsFxFMSynthChannel . . . 2080
20.35 AgsFxFMSynthChannelProcessor . . . 2082
20.36 AgsFxFMSynthRecycling . . . 2084
20.37 AgsFxSF2SynthAudio . . . 2087
20.38 AgsFxSF2SynthAudioProcessor . . . 2095
20.39 AgsFxSF2SynthAudioSignal . . . 2097
20.40 AgsFxSF2SynthChannel . . . 2099
20.41 AgsFxSF2SynthChannelProcessor . . . 2102
20.42 AgsFxSF2SynthRecycling . . . 2104

21 Fx Engine - peak and analyse 2107

21.1 AgsFxPeakAudio . . . 2107
21.2 AgsFxPeakAudioProcessor . . . 2109
21.3 AgsFxPeakAudioSignal . . . 2112
21.4 AgsFxPeakChannel . . . 2114
21.5 AgsFxPeakChannelProcessor . . . 2118
21.6 AgsFxPeakRecycling . . . 2120
21.7 AgsFxAnalyseAudio . . . 2122
21.8 AgsFxAnalyseAudioProcessor . . . 2124
21.9 AgsFxAnalyseAudioSignal . . . 2127
21.10 AgsFxAnalyseChannel . . . 2129
21.11 AgsFxAnalyseChannelProcessor . . . 2135
21.12 AgsFxAnalyseRecycling . . . 2137

22 Fx Engine - volume and envelope 2140

22.1 AgsFxVolumeAudio . . . 2140
22.2 AgsFxVolumeAudioProcessor . . . 2143
22.3 AgsFxVolumeAudioSignal . . . 2145
22.4 AgsFxVolumeChannel . . . 2147
22.5 AgsFxVolumeChannelProcessor . . . 2150
22.6 AgsFxVolumeRecycling . . . 2152
22.7 AgsFxLowPassAudio . . . 2155
22.8 AgsFxLowPassAudioProcessor . . . 2157
22.9 AgsFxLowPassAudioSignal . . . 2159
22.10 AgsFxLowPassChannel . . . 2162
22.11 AgsFxLowPassChannelProcessor . . . 2165
22.12 AgsFxLowPassRecycling . . . 2167
22.13 AgsFxHighPassAudio . . . 2169

22.14 AgsFxHighPassAudioProcessor . . . 2172
22.15 AgsFxHighPassAudioSignal . . . 2174
22.16 AgsFxHighPassChannel . . . 2176
22.17 AgsFxHighPassChannelProcessor . . . 2179
22.18 AgsFxHighPassRecycling . . . 2182
22.19 AgsFxEq10Audio . . . 2184
22.20 AgsFxEq10AudioProcessor . . . 2186
22.21 AgsFxEq10AudioSignal . . . 2189
22.22 AgsFxEq10Channel . . . 2191
22.23 AgsFxEq10ChannelProcessor . . . 2197
22.24 AgsFxEq10Recycling . . . 2199
22.25 AgsFxEnvelopeAudio . . . 2202
22.26 AgsFxEnvelopeAudioProcessor . . . 2204
22.27 AgsFxEnvelopeAudioSignal . . . 2206
22.28 AgsFxEnvelopeChannel . . . 2209
22.29 AgsFxEnvelopeChannelProcessor . . . 2213
22.30 AgsFxEnvelopeRecycling . . . 2215

23 Fx Engine - misc plugins 2218

23.1 AgsFxLadspaAudio . . . 2218
23.2 AgsFxLadspaAudioProcessor . . . 2220
23.3 AgsFxLadspaAudioSignal . . . 2223
23.4 AgsFxLadspaChannel . . . 2225
23.5 AgsFxLadspaChannelProcessor . . . 2229
23.6 AgsFxLadspaRecycling . . . 2231
23.7 AgsFxLfoAudio . . . 2233
23.8 AgsFxLfoAudioProcessor . . . 2236
23.9 AgsFxLfoAudioSignal . . . 2238
23.10 AgsFxLfoChannel . . . 2240
23.11 AgsFxLfoChannelProcessor . . . 2245
23.12 AgsFxLfoRecycling . . . 2247
23.13 AgsFxTwoPassAliaseAudio . . . 2250
23.14 AgsFxTwoPassAliaseAudioProcessor . . . 2252
23.15 AgsFxTwoPassAliaseAudioSignal . . . 2254
23.16 AgsFxTwoPassAliaseChannel . . . 2257
23.17 AgsFxTwoPassAliaseChannelProcessor . . . 2265
23.18 AgsFxTwoPassAliaseRecycling . . . 2267

24 Audio tasks - Data exchanged between Audio and GUI 2270

24.1 AgsAddAudio . . . 2270
24.2 AgsAddAudioSignal . . . 2273
24.3 AgsAddNote . . . 2276
24.4 AgsAddSoundcard . . . 2279
24.5 AgsApplyBpm . . . 2282
24.6 AgsApplyPresets . . . 2285
24.7 AgsApplySequencerLength . . . 2288
24.8 AgsApplySoundConfig . . . 2291
24.9 AgsApplySynth . . . 2294
24.10 AgsApplySF2Synth . . . 2298
24.11 AgsApplySF2MidiLocale . . . 2301
24.12 AgsApplySFZSynth . . . 2304
24.13 AgsApplySFZInstrument . . . 2308
24.14 AgsApplyTact . . . 2310
24.15 AgsCancelAudio . . . 2313
24.16 AgsCancelChannel . . . 2316
24.17 AgsClearAudioSignal . . . 2319
24.18 AgsClearBuffer . . . 2322
24.19 AgsCropNote . . . 2324
24.20 AgsExportOutput . . . 2329
24.21 AgsFreeSelection . . . 2334
24.22 AgsLinkChannel . . . 2336
24.23 AgsMoveNote . . . 2339
24.24 AgsOpenFile . . . 2345
24.25 AgsOpenSingleFile . . . 2348
24.26 AgsOpenSf2Instrument . . . 2351
24.27 AgsOpenSf2Sample . . . 2355
24.28 AgsOpenSFZFile . . . 2359
24.29 AgsOpenWave . . . 2363
24.30 AgsRemoveAudio . . . 2366
24.31 AgsRemoveAudioSignal . . . 2369
24.32 AgsRemoveNote . . . 2371
24.33 AgsRemoveSoundcard . . . 2375
24.34 AgsResetNote . . . 2377
24.35 AgsResetFxPeak . . . 2380
24.36 AgsResetFxAnalyse . . . 2383
24.37 AgsResizeAudio . . . 2386
24.38 AgsSeekSoundcard . . . 2390

24.39 AgsSetAudioChannels 2393
24.40 AgsSetBufferSize 2396
24.41 AgsSetDevice 2399
24.42 AgsSetFormat 2401
24.43 AgsSetMuted 2404
24.44 AgsSetSamplerate 2407
24.45 AgsStartAudio 2410
24.46 AgsStartChannel 2413
24.47 AgsStartSequencer 2416
24.48 AgsStartSoundcard 2418
24.49 AgsStopSequencer 2421
24.50 AgsStopSoundcard 2423
24.51 AgsStopThread 2425
24.52 AgsSwitchBufferFlag 2427
24.53 AgsTicDevice 2430
24.54 AgsTogglePatternBit 2432

III Annotation Glossary 2437

IV Tree Index 2440

25 Index 2449

Part I

Plugin - API of plugins

The plugin API contains objects to to interact with LADSPA, DSSI and LV2 plugins.

Chapter 1

Base plugin types

The base plugin types allows you to interact with plugins in a generic way.

1.1 AgsBasePlugin

AgsBasePlugin — The base plugin class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_BASE_PLUGIN_GET_OBJ_MUTEX()
GRecMutex *	ags_base_plugin_get_obj_mutex ()
gboolean	ags_base_plugin_test_flags ()
void	ags_base_plugin_set_flags ()
void	ags_base_plugin_unset_flags ()
gchar *	ags_base_plugin_get_filename ()
void	ags_base_plugin_set_filename ()
gchar *	ags_base_plugin_get_effect ()
void	ags_base_plugin_set_effect ()
guint	ags_base_plugin_get_effect_index ()
void	ags_base_plugin_set_effect_index ()
gpointer	ags_base_plugin_get_plugin_so ()
void	ags_base_plugin_set_plugin_so ()
GList *	ags_base_plugin_get_plugin_port ()
void	ags_base_plugin_set_plugin_port ()
gchar *	ags_base_plugin_get_ui_filename ()
void	ags_base_plugin_set_ui_filename ()
gchar *	ags_base_plugin_get_ui_effect ()
void	ags_base_plugin_set_ui_effect ()
guint	ags_base_plugin_get_ui_effect_index ()
void	ags_base_plugin_set_ui_effect_index ()
gpointer	ags_base_plugin_get_ui_plugin_so ()
void	ags_base_plugin_set_ui_plugin_so ()
GObject *	ags_base_plugin_get_ui_plugin ()

void	ags_base_plugin_set_ui_plugin ()
GList *	ags_base_plugin_find_filename ()
GList *	ags_base_plugin_find_effect ()
GList *	ags_base_plugin_find_ui_effect_index ()
gint	ags_base_plugin_sort_func ()
GList *	ags_base_plugin_sort ()
void	ags_base_plugin_apply_port_group_by_prefix ()
gpointer	ags_base_plugin_instantiate ()
gpointer	ags_base_plugin_instantiate_with_params ()
void	ags_base_plugin_connect_port ()
void	ags_base_plugin_activate ()
void	ags_base_plugin_deactivate ()
void	ags_base_plugin_run ()
void	ags_base_plugin_load_plugin ()
AgsBasePlugin *	ags_base_plugin_new ()
#define	AGS_BASE_PLUGIN()
#define	AGS_BASE_PLUGIN_CLASS()
#define	AGS_BASE_PLUGIN_GET_CLASS()
#define	AGS_IS_BASE_PLUGIN()
#define	AGS_IS_BASE_PLUGIN_CLASS()
GType	ags_base_plugin_flags_get_type ()
GType	ags_base_plugin_get_type ()

Properties

char *	effect	Read / Write
guint	effect-index	Read / Write
char *	filename	Read / Write
gpointer	plugin-port	Read / Write
gpointer	plugin-so	Read / Write
char *	ui-effect	Read / Write
guint	ui-effect-index	Read / Write
char *	ui-filename	Read / Write
AgsBasePlugin *	ui-plugin	Read / Write
gpointer	ui-plugin-so	Read / Write
gpointer	uuid	Read / Write

Signals

void	activate	Run Last
void	connect-port	Run Last
void	deactivate	Run Last
gpointer	instantiate	Run Last
gpointer	instantiate-with-params	Run Last
void	load-plugin	Run Last
void	run	Run Last

Types and Values

enum	AgsBasePluginFlags
#define	AGS_TYPE_BASE_PLUGIN
struct	AgsBasePlugin

struct	AgsBasePluginClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsBasePluginFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsBasePlugin
        &#x251c;&#x2500;&#x2500; AgsDssiPlugin
        &#x251c;&#x2500;&#x2500; AgsLadspaPlugin
        &#x251c;&#x2500;&#x2500; AgsLv2Plugin
        &#x2570;&#x2500;&#x2500; AgsLv2uiPlugin
```

Includes

```
#include <ags/plugin/ags_base_plugin.h>
```

Description

The AgsBasePlugin loads/unloads plugins on a abstraction level.

Functions

AGS_BASE_PLUGIN_GET_OBJ_MUTEX()

```
#define AGS_BASE_PLUGIN_GET_OBJ_MUTEX(obj) (&(((AgsBasePlugin *) obj)->obj_mutex))
```

ags_base_plugin_get_obj_mutex ()

```
GRecMutex~*
ags_base_plugin_get_obj_mutex (AgsBasePlugin *base_plugin);
```

Get object mutex.

Parameters

base_plugin	the AgsBasePlugin

Returns

the GRecMutex to lock *base_plugin*

Since: 3.1.0

ags_base_plugin_test_flags ()

```
gboolean
ags_base_plugin_test_flags (AgsBasePlugin *base_plugin,
                            guint flags);
```

Test *flags* to be set on *recall* .

Parameters

base_plugin	the AgsBasePlugin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_base_plugin_set_flags ()

```
void
ags_base_plugin_set_flags (AgsBasePlugin *base_plugin,
                           guint flags);
```

Set flags.

Parameters

base_plugin	the AgsBasePlugin	
flags	the flags	

Since: 3.0.0

ags_base_plugin_unset_flags ()

```
void
ags_base_plugin_unset_flags (AgsBasePlugin *base_plugin,
                             guint flags);
```

Unset flags.

Parameters

base_plugin	the AgsBasePlugin	
flags	the flags	

Since: 3.0.0

ags_base_plugin_get_filename ()

```
gchar~*
ags_base_plugin_get_filename (AgsBasePlugin *base_plugin);
```

Get filename.

Parameters

base_plugin	the AgsBasePlugin

Returns

the filename

Since: 3.1.0

ags_base_plugin_set_filename ()

```
void
ags_base_plugin_set_filename (AgsBasePlugin *base_plugin,
                              gchar *filename);
```

Set filename.

Parameters

base_plugin	the AgsBasePlugin
filename	the filename

Since: 3.1.0

ags_base_plugin_get_effect ()

```
gchar~*
ags_base_plugin_get_effect (AgsBasePlugin *base_plugin);
```

Get effect.

Parameters

base_plugin	the AgsBasePlugin

Returns

the effect

Since: 3.1.0

ags_base_plugin_set_effect ()

```
void
ags_base_plugin_set_effect (AgsBasePlugin *base_plugin,
                            gchar *effect);
```

Set effect.

Parameters

base_plugin	the AgsBasePlugin	
effect	the effect	

Since: 3.1.0

ags_base_plugin_get_effect_index ()

```
guint
ags_base_plugin_get_effect_index (AgsBasePlugin *base_plugin);
```

Get effect index.

Parameters

base_plugin	the AgsBasePlugin	

Returns

the effect index

Since: 3.1.0

ags_base_plugin_set_effect_index ()

```
void
ags_base_plugin_set_effect_index (AgsBasePlugin *base_plugin,
                                  guint effect_index);
```

Set effect index.

Parameters

base_plugin	the AgsBasePlugin	
effect_index	the effect index	

Since: 3.1.0

ags_base_plugin_get_plugin_so ()

```
gpointer
ags_base_plugin_get_plugin_so (AgsBasePlugin *base_plugin);
```

Get plugin so.

Parameters

base_plugin	the AgsBasePlugin	

Returns

the plugin so

Since: 3.1.0

ags_base_plugin_set_plugin_so ()

```
void
ags_base_plugin_set_plugin_so (AgsBasePlugin *base_plugin,
                               gpointer plugin_so);
```

Set plugin_so.

Parameters

base_plugin	the AgsBasePlugin	
plugin_so	the plugin so	

Since: 3.1.0

ags_base_plugin_get_plugin_port ()

```
GList~*
ags_base_plugin_get_plugin_port (AgsBasePlugin *base_plugin);
```

ags_base_plugin_set_plugin_port ()

```
void
ags_base_plugin_set_plugin_port (AgsBasePlugin *base_plugin,
                                 GList *plugin_port);
```

Set recall dependency by replacing existing.

Parameters

base_plugin	the AgsBasePlugin	
plugin_port	the GList containing AgsPluginPort.	*[element-type AgsAudio.PluginPort][transfer full]*

Since: 3.1.0

ags_base_plugin_get_ui_filename ()

```
gchar~*
ags_base_plugin_get_ui_filename (AgsBasePlugin *base_plugin);
```

Get UI filename.

Parameters

base_plugin	the AgsBasePlugin

Returns

the UI filename

Since: 3.1.0

ags_base_plugin_set_ui_filename ()

```
void
ags_base_plugin_set_ui_filename (AgsBasePlugin *base_plugin,
                                 gchar *ui_filename);
```

Set UI filename.

Parameters

base_plugin	the AgsBasePlugin
ui_filename	the UI filename

Since: 3.1.0

ags_base_plugin_get_ui_effect ()

```
gchar~*
ags_base_plugin_get_ui_effect (AgsBasePlugin *base_plugin);
```

Get UI effect.

Parameters

base_plugin	the AgsBasePlugin

Returns

the UI effect

Since: 3.1.0

ags_base_plugin_set_ui_effect ()

```
void
ags_base_plugin_set_ui_effect (AgsBasePlugin *base_plugin,
                               gchar *ui_effect);
```

Set UI effect.

Parameters

base_plugin	the AgsBasePlugin	
ui_effect	the UI effect	

Since: 3.1.0

ags_base_plugin_get_ui_effect_index ()

```
guint
ags_base_plugin_get_ui_effect_index (AgsBasePlugin *base_plugin);
```

Get UI effect index.

Parameters

base_plugin	the AgsBasePlugin	

Returns

the UI effect index

Since: 3.1.0

ags_base_plugin_set_ui_effect_index ()

```
void
ags_base_plugin_set_ui_effect_index (AgsBasePlugin *base_plugin,
                                     guint ui_effect_index);
```

Set UI effect index.

Parameters

base_plugin	the AgsBasePlugin	
ui_effect_index	the UI effect index	

Since: 3.1.0

ags_base_plugin_get_ui_plugin_so ()

```
gpointer
ags_base_plugin_get_ui_plugin_so (AgsBasePlugin *base_plugin);
```

Get plugin so.

Parameters

base_plugin	the AgsBasePlugin	

Returns

the plugin so

Since: 3.1.0

ags_base_plugin_set_ui_plugin_so ()

```
void
ags_base_plugin_set_ui_plugin_so (AgsBasePlugin *base_plugin,
                                  gpointer ui_plugin_so);
```

Set ui_plugin_so.

Parameters

base_plugin	the AgsBasePlugin	
ui_plugin_so	the plugin so	

Since: 3.1.0

ags_base_plugin_get_ui_plugin ()

```
GObject~*
ags_base_plugin_get_ui_plugin (AgsBasePlugin *base_plugin);
```

ags_base_plugin_set_ui_plugin ()

```
void
ags_base_plugin_set_ui_plugin (AgsBasePlugin *base_plugin,
                               GObject *ui_plugin);
```

ags_base_plugin_find_filename ()

```
GList~*
ags_base_plugin_find_filename (GList *base_plugin,
                               gchar *filename);
```

Find filename in `base_plugin` GList of AgsBasePlugin.

Parameters

base_plugin	the GList containing AgsBasePlugin.	*[element-type AgsAudio.BasePlugin]*
filename	the filename as string	

Returns

the next matching GList.

[element-type AgsAudio.BasePlugin][transfer none]

Since: 3.0.0

ags_base_plugin_find_effect ()

```
GList~*
ags_base_plugin_find_effect (GList *base_plugin,
                             gchar *filename,
                             gchar *effect);
```

Find *filename* and *effect* in *base_plugin* GList of AgsBasePlugin.

Parameters

base_plugin	the GList containing AgsBasePlugin.	*[element-type AgsAudio.BasePlugin]*
filename	the filename as string	
effect	the effect as string	

Returns

the next matching GList.

[element-type AgsAudio.BasePlugin][transfer none]

Since: 3.0.0

ags_base_plugin_find_ui_effect_index ()

```
GList~*
ags_base_plugin_find_ui_effect_index (GList *base_plugin,
                                      gchar *ui_filename,
                                      guint ui_effect_index);
```

Find *ui_filename* and *ui_effect_index* in *base_plugin* GList of AgsBasePlugin.

Parameters

base_plugin	the GList containing AgsBasePlugin.	*[element-type AgsAudio.BasePlugin]*
ui_filename	the UI filename as string	
ui_effect_index	the UI effect index	

Returns

the next matching GList.

[element-type AgsAudio.BasePlugin][transfer none]

Since: 3.0.0

ags_base_plugin_sort_func ()

```
gint
ags_base_plugin_sort_func (gpointer a,
                           gpointer b);
```

Sort function.

Parameters

a	a AgsBasePlugin	
b	an other AgsBasePlugin	

Returns

a value smaller to, equal to or greater to 0 like strcmp

Since: 3.0.0

ags_base_plugin_sort ()

```
GList~*
ags_base_plugin_sort (GList *base_plugin);
```

Sort *base_plugin* alphabetically.

Parameters

base_plugin	the GList containing AgsBasePlugin.	*[element-type AgsAudio.BasePlugin][transfer none]*

Returns

the sorted GList.

[element-type AgsAudio.BasePlugin][transfer none]

Since: 3.0.0

ags_base_plugin_apply_port_group_by_prefix ()

```
void
ags_base_plugin_apply_port_group_by_prefix
                                (AgsBasePlugin *base_plugin);
```

ags_base_plugin_instantiate ()

```
gpointer
ags_base_plugin_instantiate (AgsBasePlugin *base_plugin,
                             guint samplerate,
                             guint buffer_size);
```

Instantiate the plugin

Parameters

base_plugin	the AgsBasePlugin	
samplerate	the samplerate	
buffer_size	the fixed buffer size	

Returns

the new plugin instance handle

Since: 3.0.0

ags_base_plugin_instantiate_with_params ()

```
gpointer
ags_base_plugin_instantiate_with_params
                                (AgsBasePlugin *base_plugin,
                                 guint *n_params,
                                 gchar ***parameter_name,
                                 GValue **value);
```

Instantiate the plugin

Parameters

base_plugin	the AgsBasePlugin	
n_params	guint pointer to parameter count	
parameter_name	string vector containing parameter names	
value	the GValue array	

Returns

the new plugin instance handle

Since: 3.0.0

ags_base_plugin_connect_port ()

```
void
ags_base_plugin_connect_port (AgsBasePlugin *base_plugin,
                              gpointer plugin_handle,
                              guint port_index,
                              gpointer data_location);
```

Connect a plugin instance.

Parameters

base_plugin	the AgsBasePlugin	
plugin_handle	the plugin instance handle	
port_index	the port's index to connect	
data_location	the data location to connect	

Since: 3.0.0

ags_base_plugin_activate ()

```
void
ags_base_plugin_activate (AgsBasePlugin *base_plugin,
                          gpointer plugin_handle);
```

Activate a plugin instance

Parameters

base_plugin	the AgsBasePlugin	
plugin_handle	the plugin instance handle	

Since: 3.0.0

ags_base_plugin_deactivate ()

```
void
ags_base_plugin_deactivate (AgsBasePlugin *base_plugin,
                            gpointer plugin_handle);
```

Deactivat a plugin instance

Parameters

base_plugin	the AgsBasePlugin	
plugin_handle	the plugin instance handle	

Since: 3.0.0

ags_base_plugin_run ()

```
void
ags_base_plugin_run (AgsBasePlugin *base_plugin,
                     gpointer plugin_handle,
                     snd_seq_event_t *seq_event,
                     guint frame_count);
```

Deactivat a plugin instance

Parameters

base_plugin	the AgsBasePlugin	
plugin_handle	the plugin instance handle	
seq_event	the alsa sequencer events.	*[type gpointer][transfer none]*
frame_count	the frame counts	

Since: 3.0.0

ags_base_plugin_load_plugin ()

```
void
ags_base_plugin_load_plugin (AgsBasePlugin *base_plugin);
```

Load the plugin

Parameters

base_plugin	the AgsBasePlugin

Since: 3.0.0

ags_base_plugin_new ()

```
AgsBasePlugin~*
ags_base_plugin_new (gchar *filename,
                     gchar *effect,
                     guint effect_index);
```

Creates an AgsBasePlugin

Parameters

filename	the plugin .so
effect	the effect's string representation
effect_index	the effect's index

Returns

a new AgsBasePlugin

Since: 3.0.0

AGS_BASE_PLUGIN()

```
#define AGS_BASE_PLUGIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_BASE_PLUGIN, AgsBasePlugin))
```

AGS_BASE_PLUGIN_CLASS()

```
#define AGS_BASE_PLUGIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_BASE_PLUGIN, AgsBasePluginClass))
```

AGS_BASE_PLUGIN_GET_CLASS()

```
#define AGS_BASE_PLUGIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_BASE_PLUGIN, AgsBasePluginClass))
```

AGS_IS_BASE_PLUGIN()

```
#define AGS_IS_BASE_PLUGIN(obj)                (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_BASE_PLUGIN))
```

AGS_IS_BASE_PLUGIN_CLASS()

```
#define AGS_IS_BASE_PLUGIN_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_BASE_PLUGIN))
```

ags_base_plugin_flags_get_type ()

```
GType
ags_base_plugin_flags_get_type ();
```

ags_base_plugin_get_type ()

```
GType
ags_base_plugin_get_type (void);
```

Types and Values

enum AgsBasePluginFlags

Enum values to control the behavior or indicate internal state of AgsBasePlugin by enable/disable as flags.

Members

AGS_BASE_PLUGIN_IS_INSTRUMENT	plugin is instrument

AGS_TYPE_BASE_PLUGIN

```
#define AGS_TYPE_BASE_PLUGIN                   (ags_base_plugin_get_type())
```

struct AgsBasePlugin

```
struct AgsBasePlugin;
```

struct AgsBasePluginClass

```
struct AgsBasePluginClass {
  GObjectClass gobject;

  gpointer (*instantiate)(AgsBasePlugin *base_plugin,
        guint samplerate, guint buffer_size);

  gpointer (*instantiate_with_params)(AgsBasePlugin *base_plugin,
              guint *n_params,
              gchar ***parameter_name,
              GValue **value);

  void (*connect_port)(AgsBasePlugin *base_plugin, gpointer plugin_handle, guint port_index ↩
      , gpointer data_location);

  void (*activate)(AgsBasePlugin *base_plugin, gpointer plugin_handle);
  void (*deactivate)(AgsBasePlugin *base_plugin, gpointer plugin_handle);

  void (*run)(AgsBasePlugin *base_plugin,
        gpointer plugin_handle,
        snd_seq_event_t *seq_event,
        guint frame_count);

  void (*load_plugin)(AgsBasePlugin *base_plugin);
};
```

Property Details

The "effect" property

```
  "effect"                    char~*
```

The assigned effect.

Owner: AgsBasePlugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "effect-index" property

```
  "effect-index"              guint
```

The assigned effect-index.

Owner: AgsBasePlugin

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsBasePlugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "plugin-port" property

```
"plugin-port"               gpointer
```

The assigned GList containing AgsPluginPort

[transfer full]

Owner: AgsBasePlugin

Flags: Read / Write

Since: 3.0.0

The "plugin-so" property

```
"plugin-so"                 gpointer
```

The assigned plugin.so

Owner: AgsBasePlugin

Flags: Read / Write

Since: 3.0.0

The "ui-effect" property

```
"ui-effect"                 char~*
```

The assigned ui-effect.

Owner: AgsBasePlugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ui-effect-index" property

```
"ui-effect-index"           guint
```

The assigned ui-effect-index.

Owner: AgsBasePlugin

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "ui-filename" property

```
  "ui-filename"                char~*
```

The assigned UI filename.

Owner: AgsBasePlugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ui-plugin" property

```
  "ui-plugin"                  AgsBasePlugin~*
```

The assigned ui-plugin.

Owner: AgsBasePlugin

Flags: Read / Write

Since: 3.0.0

The "ui-plugin-so" property

```
  "ui-plugin-so"               gpointer
```

The assigned ui_plugin.so

Owner: AgsBasePlugin

Flags: Read / Write

Since: 3.0.0

The "uuid" property

```
  "uuid"                       gpointer
```

The assigned AgsUUID

Owner: AgsBasePlugin

Flags: Read / Write

Since: 3.0.0

Signal Details

The "activate" signal

```
void
user_function (AgsBasePlugin *base_plugin,
               gpointer       plugin_handle,
               gpointer       user_data)
```

The ::activate signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to activate	
plugin_handle	the plugin handle	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "connect-port" signal

```
void
user_function (AgsBasePlugin *base_plugin,
               gpointer       plugin_handle,
               guint          port_index,
               gpointer       data_location,
               gpointer       user_data)
```

The ::connect-port signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to connect-port	
plugin_handle	the plugin handle	
port_index	the port index	
data_location	the data location	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "deactivate" signal

```
void
user_function (AgsBasePlugin *base_plugin,
               gpointer       plugin_handle,
               gpointer       user_data)
```

The ::deactivate signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to deactivate	
plugin_handle	the plugin handle	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "instantiate" signal

```
gpointer
user_function (AgsBasePlugin *base_plugin,
               guint          samplerate,
               guint          buffer_size,
               gpointer       user_data)
```

The ::instantiate signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to instantiate	
samplerate	the samplerate	
buffer_size	the fixed buffer size	
user_data	user data set when the signal handler was connected.	

Returns

the new plugin instance

Flags: Run Last

Since: 3.0.0

The "instantiate-with-params" signal

```
gpointer
user_function (AgsBasePlugin *base_plugin,
               gpointer       n_params,
               gpointer       parameter_name,
               gpointer       value,
               gpointer       user_data)
```

The ::instantiate-with-params signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to instantiate	
n_params	pointer to array length	
parameter_name	parameter name string vector	
value	the GValue array	
user_data	user data set when the signal handler was connected.	

Returns

the new plugin instance

Flags: Run Last

Since: 3.0.0

The "load-plugin" signal

```
void
user_function (AgsBasePlugin *base_plugin,
               gpointer       user_data)
```

The ::load-plugin signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to load_plugin	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run" signal

```
void
user_function (AgsBasePlugin *base_plugin,
               gpointer       plugin_handle,
               gpointer       seq_event,
               guint          frame_count,
               gpointer       user_data)
```

The ::run signal creates a new instance of plugin.

Parameters

base_plugin	the plugin to run	
plugin_handle	the plugin handle	
seq_event	the MIDI data	
frame_count	the frame count	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

1.2 ags_plugin_stock

ags_plugin_stock —

Stability Level

Stable, unless otherwise indicated

Types and Values

#define	AGS_PLUGIN_NAME_VOLUME

Description

Functions

Types and Values

AGS_PLUGIN_NAME_VOLUME

```
#define AGS_PLUGIN_NAME_VOLUME "ags-volume\0"
```

1.3 AgsPluginPort

AgsPluginPort — The plugin port class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PLUGIN_PORT_GET_OBJ_MUTEX()
GRecMutex *	ags_plugin_port_get_obj_mutex ()
gboolean	ags_plugin_port_test_flags ()
void	ags_plugin_port_set_flags ()
void	ags_plugin_port_unset_flags ()
guint	ags_plugin_port_get_port_index ()
void	ags_plugin_port_set_port_index ()
gchar *	ags_plugin_port_get_port_name ()
void	ags_plugin_port_set_port_name ()
gchar *	ags_plugin_port_get_port_symbol ()
void	ags_plugin_port_set_port_symbol ()
gint	ags_plugin_port_get_scale_steps ()
void	ags_plugin_port_set_scale_steps ()
gchar **	ags_plugin_port_get_scale_point ()
void	ags_plugin_port_set_scale_point ()
gdouble *	ags_plugin_port_get_scale_value ()
void	ags_plugin_port_set_scale_value ()
GValue *	ags_plugin_port_get_lower_value ()
void	ags_plugin_port_set_lower_value ()
GValue *	ags_plugin_port_get_upper_value ()
void	ags_plugin_port_set_upper_value ()
GValue *	ags_plugin_port_get_default_value ()
void	ags_plugin_port_set_default_value ()
GList *	ags_plugin_port_find_symbol ()
GList *	ags_plugin_port_find_port_index ()
AgsPluginPort *	ags_plugin_port_new ()

#define	AGS_IS_PLUGIN_PORT()
#define	AGS_IS_PLUGIN_PORT_CLASS()
#define	AGS_PLUGIN_PORT()
#define	AGS_PLUGIN_PORT_CLASS()
#define	AGS_PLUGIN_PORT_GET_CLASS()
GType	ags_plugin_port_flags_get_type ()
GType	ags_plugin_port_get_type ()

Properties

gpointer	default-value	Read / Write
gpointer	lower-value	Read / Write
guint	port-index	Read / Write
char *	port-name	Read / Write
char *	port-symbol	Read / Write
gpointer	scale-point	Read
int	scale-steps	Read / Write
gpointer	scale-value	Read
gpointer	upper-value	Read / Write

Types and Values

enum	AgsPluginPortFlags
#define	AGS_TYPE_PLUGIN_PORT
struct	AgsPluginPort
struct	AgsPluginPortClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsPluginPortFlags
GObject
&#x2570;&#x2500;&#x2500; AgsPluginPort
```

Includes

```
#include <ags/plugin/ags_plugin_port.h>
```

Description

The AgsPluginPort describes a plugin's port. For real ports used by the processing tree please take a look at AgsPort.

Functions

AGS_PLUGIN_PORT_GET_OBJ_MUTEX()

```
#define AGS_PLUGIN_PORT_GET_OBJ_MUTEX(obj) (&(((AgsPluginPort *) obj)->obj_mutex))
```

ags_plugin_port_get_obj_mutex ()

```
GRecMutex~*
ags_plugin_port_get_obj_mutex (AgsPluginPort *plugin_port);
```

Get object mutex.

Parameters

plugin_port	the AgsPluginPort

Returns

the GRecMutex to lock *plugin_port*

Since: 3.1.0

ags_plugin_port_test_flags ()

```
gboolean
ags_plugin_port_test_flags (AgsPluginPort *plugin_port,
                            guint flags);
```

Test *flags* to be set on *plugin_port* .

Parameters

plugin_port	the AgsPluginPort
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_plugin_port_set_flags ()

```
void
ags_plugin_port_set_flags (AgsPluginPort *plugin_port,
                           guint flags);
```

Set *flags* on *plugin_port* .

Parameters

plugin_port	the AgsPluginPort
flags	the flags

Since: 3.0.0

ags_plugin_port_unset_flags ()

```
void
ags_plugin_port_unset_flags (AgsPluginPort *plugin_port,
                             guint flags);
```

Unset *flags* on *plugin_port* .

Parameters

plugin_port	the AgsPluginPort	
flags	the flags	

Since: 3.0.0

ags_plugin_port_get_port_index ()

```
guint
ags_plugin_port_get_port_index (AgsPluginPort *plugin_port);
```

Get port index.

Parameters

plugin_port	the AgsPluginPort	

Returns

the port index

Since: 3.1.0

ags_plugin_port_set_port_index ()

```
void
ags_plugin_port_set_port_index (AgsPluginPort *plugin_port,
                                guint port_index);
```

Set port index.

Parameters

plugin_port	the AgsPluginPort	
port_index	the port index	

Since: 3.1.0

ags_plugin_port_get_port_name ()

```
gchar~*
ags_plugin_port_get_port_name (AgsPluginPort *plugin_port);
```

Get port name.

Parameters

plugin_port	the AgsPluginPort

Returns

the port name

Since: 3.1.0

ags_plugin_port_set_port_name ()

```
void
ags_plugin_port_set_port_name (AgsPluginPort *plugin_port,
                               gchar *port_name);
```

Set port name.

Parameters

plugin_port	the AgsPluginPort
port_name	the port name

Since: 3.1.0

ags_plugin_port_get_port_symbol ()

```
gchar~*
ags_plugin_port_get_port_symbol (AgsPluginPort *plugin_port);
```

Get port symbol.

Parameters

plugin_port	the AgsPluginPort

Returns

the port symbol

Since: 3.1.0

ags_plugin_port_set_port_symbol ()

```
void
ags_plugin_port_set_port_symbol (AgsPluginPort *plugin_port,
                                 gchar *port_symbol);
```

Set port symbol.

Parameters

plugin_port	the AgsPluginPort	
port_symbol	the port symbol	

Since: 3.1.0

ags_plugin_port_get_scale_steps ()

```
gint
ags_plugin_port_get_scale_steps (AgsPluginPort *plugin_port);
```

Get scale steps.

Parameters

plugin_port	the AgsPluginPort	

Returns

the scale steps

Since: 3.1.0

ags_plugin_port_set_scale_steps ()

```
void
ags_plugin_port_set_scale_steps (AgsPluginPort *plugin_port,
                                 gint scale_steps);
```

Set scale steps.

Parameters

plugin_port	the AgsPluginPort	
scale_steps	the scale steps	

Since: 3.1.0

ags_plugin_port_get_scale_point ()

```
gchar~**
ags_plugin_port_get_scale_point (AgsPluginPort *plugin_port);
```

Get scale point.

Parameters

plugin_port	the AgsPluginPort	

Returns

the scale point

Since: 3.1.0

ags_plugin_port_set_scale_point ()

```
void
ags_plugin_port_set_scale_point (AgsPluginPort *plugin_port,
                                 gchar **scale_point);
```

Set scale point.

Parameters

plugin_port	the AgsPluginPort	
scale_point	the scale point	

Since: 3.1.0

ags_plugin_port_get_scale_value ()

```
gdouble~*
ags_plugin_port_get_scale_value (AgsPluginPort *plugin_port);
```

Get scale value.

Parameters

plugin_port	the AgsPluginPort	

Returns

the scale value

Since: 3.1.0

ags_plugin_port_set_scale_value ()

```
void
ags_plugin_port_set_scale_value (AgsPluginPort *plugin_port,
                                 gdouble *scale_value);
```

Set scale value.

Parameters

plugin_port	the AgsPluginPort	
scale_value	the scale value	

Since: 3.1.0

ags_plugin_port_get_lower_value ()

```
GValue~*
ags_plugin_port_get_lower_value (AgsPluginPort *plugin_port);
```

Get lower value.

Parameters

plugin_port	the AgsPluginPort	

Returns

the lower value

Since: 3.1.0

ags_plugin_port_set_lower_value ()

```
void
ags_plugin_port_set_lower_value (AgsPluginPort *plugin_port,
                                 GValue *lower_value);
```

Set lower value.

Parameters

plugin_port	the AgsPluginPort	
lower_value	the lower value	

Since: 3.1.0

ags_plugin_port_get_upper_value ()

```
GValue~*
ags_plugin_port_get_upper_value (AgsPluginPort *plugin_port);
```

Get upper value.

Parameters

plugin_port	the AgsPluginPort	

Returns

the upper value

Since: 3.1.0

ags_plugin_port_set_upper_value ()

```
void
ags_plugin_port_set_upper_value (AgsPluginPort *plugin_port,
                                 GValue *upper_value);
```

Set upper value.

Parameters

plugin_port	the AgsPluginPort	
upper_value	the upper value	

Since: 3.1.0

ags_plugin_port_get_default_value ()

```
GValue~*
ags_plugin_port_get_default_value (AgsPluginPort *plugin_port);
```

Get default value.

Parameters

plugin_port	the AgsPluginPort	

Returns

the default value

Since: 3.1.0

ags_plugin_port_set_default_value ()

```
void
ags_plugin_port_set_default_value (AgsPluginPort *plugin_port,
                                   GValue *default_value);
```

Set default value.

Parameters

plugin_port	the AgsPluginPort	
default_value	the default value	

Since: 3.1.0

ags_plugin_port_find_symbol ()

```
GList~*
ags_plugin_port_find_symbol (GList *plugin_port,
                             gchar *port_symbol);
```

Find *port_symbol* within *plugin_port* .

Parameters

plugin_port	the GList containing AgsPluginPort.	*[element-type AgsAudio.PluginPort][transfer none]*
port_symbol	the port symbol	

Returns

the matching GList containing AgsPluginPort.

[element-type AgsAudio.PluginPort][transfer none]

Since: 3.0.0

ags_plugin_port_find_port_index ()

```
GList~*
ags_plugin_port_find_port_index (GList *plugin_port,
                                 guint port_index);
```

Find `port_index` within `plugin_port` .

Parameters

plugin_port	the GList containing AgsPluginPort.	*[element-type AgsAudio.PluginPort][transfer none]*
port_index	the port index	

Returns

the matching GList containing AgsPluginPort.

[element-type AgsAudio.PluginPort][transfer none]

Since: 3.0.0

ags_plugin_port_new ()

```
AgsPluginPort~*
ags_plugin_port_new ();
```

Creates an AgsPluginPort

Returns

a new AgsPluginPort

Since: 3.0.0

AGS_IS_PLUGIN_PORT()

```
#define AGS_IS_PLUGIN_PORT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PLUGIN_PORT))
```

AGS_IS_PLUGIN_PORT_CLASS()

```
#define AGS_IS_PLUGIN_PORT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PLUGIN_PORT))
```

AGS_PLUGIN_PORT()

```
#define AGS_PLUGIN_PORT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PLUGIN_PORT, AgsPluginPort))
```

AGS_PLUGIN_PORT_CLASS()

```
#define AGS_PLUGIN_PORT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_PLUGIN_PORT, AgsPluginPortClass))
```

AGS_PLUGIN_PORT_GET_CLASS()

```
#define AGS_PLUGIN_PORT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_PLUGIN_PORT, AgsPluginPortClass))
```

ags_plugin_port_flags_get_type ()

```
GType
ags_plugin_port_flags_get_type ();
```

ags_plugin_port_get_type ()

```
GType
ags_plugin_port_get_type (void);
```

Types and Values

enum AgsPluginPortFlags

Common port attributes.

Members

AGS_PLUGIN_PORT_ATOM	atom port
AGS_PLUGIN_PORT_AUDIO	audio port
AGS_PLUGIN_PORT_CONTROL	control port
AGS_PLUGIN_PORT_MIDI	MIDI port

AGS_PLUGIN_PORT_EVENT	event port
AGS_PLUGIN_PORT_OUTPUT	is output
AGS_PLUGIN_PORT_INPUT	is input
AGS_PLUGIN_PORT_TOGGLED	toggle control
AGS_PLUGIN_PORT_ENUMERATION	enumerated
AGS_PLUGIN_PORT_LOGARITHMIC	logarithmic
AGS_PLUGIN_PORT_INTEGER	integer
AGS_PLUGIN_PORT_SAMPLERATE	samplerate
AGS_PLUGIN_PORT_BOUNDED_BELOW	bounded below
AGS_PLUGIN_PORT_BOUNDED_ABOVE	bounded above
AGS_PLUGIN_PORT_UI_NOTIFICATION	ui notification
AGS_PLUGIN_PORT_HIDDEN	hidden port

AGS_TYPE_PLUGIN_PORT

```
#define AGS_TYPE_PLUGIN_PORT                (ags_plugin_port_get_type())
```

struct AgsPluginPort

```
struct AgsPluginPort;
```

struct AgsPluginPortClass

```
struct AgsPluginPortClass {
  GObjectClass gobject;
};
```

Property Details

The "default-value" property

```
  "default-value"            gpointer
```

The default value.

Owner: AgsPluginPort

Flags: Read / Write

Since: 3.0.0

The "lower-value" property

```
"lower-value"                gpointer
```

The lower value.

Owner: AgsPluginPort

Flags: Read / Write

Since: 3.0.0

The "port-index" property

```
"port-index"                 guint
```

The assigned port-index.

Owner: AgsPluginPort

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "port-name" property

```
"port-name"                  char~*
```

The port's name.

Owner: AgsPluginPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "port-symbol" property

```
"port-symbol"                char~*
```

The port's symbol.

Owner: AgsPluginPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "scale-point" property

```
"scale-point"                gpointer
```

The scale point string vector.

Owner: AgsPluginPort

Flags: Read

Since: 3.0.0

The "scale-steps" property

```
"scale-steps"                int
```

The number of scale steps.

Owner: AgsPluginPort

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "scale-value" property

```
"scale-value"                gpointer
```

The scale value array.

Owner: AgsPluginPort

Flags: Read

Since: 3.0.0

The "upper-value" property

```
"upper-value"                gpointer
```

The upper value.

Owner: AgsPluginPort

Flags: Read / Write

Since: 3.0.0

Chapter 2

LADSPA plugins

The LADSPA plugin objects.

2.1 AgsLadspaConversion

AgsLadspaConversion — Conversion of values

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_ladspa_conversion_test_flags ()
void	ags_ladspa_conversion_set_flags ()
void	ags_ladspa_conversion_unset_flags ()
AgsLadspaConversion *	ags_ladspa_conversion_new ()
#define	AGS_IS_LADSPA_CONVERSION()
#define	AGS_IS_LADSPA_CONVERSION_CLASS()
#define	AGS_LADSPA_CONVERSION()
#define	AGS_LADSPA_CONVERSION_CLASS()
#define	AGS_LADSPA_CONVERSION_GET_CLASS()
GType	ags_ladspa_conversion_flags_get_type ()
GType	ags_ladspa_conversion_get_type ()

Properties

double	lower	Read / Write
guint	samplerate	Read / Write
double	step-count	Read / Write
double	upper	Read / Write

Types and Values

#define	AGS_LADSPA_CONVERSION_DEFAULT_LOWER
#define	AGS_LADSPA_CONVERSION_DEFAULT_UPPER
#define	AGS_LADSPA_CONVERSION_DEFAULT_STEP_COUNT
enum	AgsLadspaConversionFlags
#define	AGS_TYPE_LADSPA_CONVERSION
struct	AgsLadspaConversion
struct	AgsLadspaConversionClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsLadspaConversionFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsConversion
        &#x2570;&#x2500;&#x2500; AgsLadspaConversion
```

Includes

```
#include <ags/plugin/ags_ladspa_conversion.h>
```

Description

The AgsLadspaConversion converts values.

Functions

ags_ladspa_conversion_test_flags ()

```
gboolean
ags_ladspa_conversion_test_flags (AgsLadspaConversion *ladspa_conversion,
                                  guint flags);
```

Test *flags* to be set on *recall* .

Parameters

ladspa_conversion	the AgsLadspaConversion	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_ladspa_conversion_set_flags ()

```
void
ags_ladspa_conversion_set_flags (AgsLadspaConversion *ladspa_conversion,
                                 guint flags);
```

Set flags.

Parameters

ladspa_conversion	the AgsLadspaConversion	
flags	the flags	

Since: 3.0.0

ags_ladspa_conversion_unset_flags ()

```
void
ags_ladspa_conversion_unset_flags (AgsLadspaConversion *ladspa_conversion,
                                   guint flags);
```

Unset flags.

Parameters

ladspa_conversion	the AgsLadspaConversion	
flags	the flags	

Since: 3.0.0

ags_ladspa_conversion_new ()

```
AgsLadspaConversion~*
ags_ladspa_conversion_new ();
```

Create a new instance of AgsLadspaConversion.

Returns

the new AgsLadspaConversion.

Since: 3.0.0

AGS_IS_LADSPA_CONVERSION()

```
#define AGS_IS_LADSPA_CONVERSION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CONVERSION))
```

AGS_IS_LADSPA_CONVERSION_CLASS()

```
#define AGS_IS_LADSPA_CONVERSION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CONVERSION))
```

AGS_LADSPA_CONVERSION()

```
#define AGS_LADSPA_CONVERSION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CONVERSION, AgsLadspaConversion))
```

AGS_LADSPA_CONVERSION_CLASS()

```
#define AGS_LADSPA_CONVERSION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CONVERSION, AgsLadspaConversionClass))
```

AGS_LADSPA_CONVERSION_GET_CLASS()

```
#define AGS_LADSPA_CONVERSION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_CONVERSION, AgsLadspaConversionClass))
```

ags_ladspa_conversion_flags_get_type ()

```
GType
ags_ladspa_conversion_flags_get_type ();
```

ags_ladspa_conversion_get_type ()

```
GType
ags_ladspa_conversion_get_type (void);
```

Types and Values

AGS_LADSPA_CONVERSION_DEFAULT_LOWER

```
#define AGS_LADSPA_CONVERSION_DEFAULT_LOWER (0.00001)
```

AGS_LADSPA_CONVERSION_DEFAULT_UPPER

```
#define AGS_LADSPA_CONVERSION_DEFAULT_UPPER (1.0)
```

AGS_LADSPA_CONVERSION_DEFAULT_STEP_COUNT

```
#define AGS_LADSPA_CONVERSION_DEFAULT_STEP_COUNT (1025.0)
```

enum AgsLadspaConversionFlags

Enum values to control the behavior or indicate internal state of AgsLadspaConversion by enable/disable as flags.

Members

AGS_LADSPA_CONVERSION_SAMPLERATE	samplerate
AGS_LADSPA_CONVERSION_BOUNDED_BELOW	bounded be-low
AGS_LADSPA_CONVERSION_BOUNDED_ABOVE	bounded above
AGS_LADSPA_CONVERSION_LOGARITHMIC	logarithmic

AGS_TYPE_LADSPA_CONVERSION

```
#define AGS_TYPE_LADSPA_CONVERSION                (ags_ladspa_conversion_get_type())
```

struct AgsLadspaConversion

```
struct AgsLadspaConversion;
```

struct AgsLadspaConversionClass

```
struct AgsLadspaConversionClass {
  AgsConversionClass conversion;
};
```

Property Details

The "lower" property

```
  "lower"                    double
```

The lower to be used.

Owner: AgsLadspaConversion

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "samplerate" property

```
  "samplerate"               guint
```

The samplerate to be used.

Owner: AgsLadspaConversion

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "step-count" property

```
"step-count"                double
```

The step count to be used.

Owner: AgsLadspaConversion

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "upper" property

```
"upper"                     double
```

The upper to be used.

Owner: AgsLadspaConversion

Flags: Read / Write

Default value: 0

Since: 3.0.0

2.2 AgsLadspaManager

AgsLadspaManager — Singleton pattern to organize LADSPA

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LADSPA_MANAGER_GET_OBJ_MUTEX()
gchar **	ags_ladspa_manager_get_default_path ()
void	ags_ladspa_manager_set_default_path ()
gchar **	ags_ladspa_manager_get_filenames ()
AgsLadspaPlugin *	ags_ladspa_manager_find_ladspa_plugin ()
AgsLadspaPlugin *	ags_ladspa_manager_find_ladspa_plugin_with_fallback ()
void	ags_ladspa_manager_load_blacklist ()
void	ags_ladspa_manager_load_file ()
void	ags_ladspa_manager_load_default_directory ()
GList *	ags_ladspa_manager_get_ladspa_plugin ()
AgsLadspaManager *	ags_ladspa_manager_get_instance ()
AgsLadspaManager *	ags_ladspa_manager_new ()
#define	AGS_IS_LADSPA_MANAGER()
#define	AGS_IS_LADSPA_MANAGER_CLASS()
#define	AGS_LADSPA_MANAGER()
#define	AGS_LADSPA_MANAGER_CLASS()

#define	AGS_LADSPA_MANAGER_GET_CLASS()
GType	ags_ladspa_manager_get_type ()

Types and Values

#define	AGS_TYPE_LADSPA_MANAGER
struct	AgsLadspaManager
struct	AgsLadspaManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsLadspaManager
```

Includes

```
#include <ags/plugin/ags_ladspa_manager.h>
```

Description

The AgsLadspaManager loads/unloads LADSPA plugins.

Functions

AGS_LADSPA_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LADSPA_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLadspaManager *) obj)->obj_mutex))
```

ags_ladspa_manager_get_default_path ()

```
gchar~**
ags_ladspa_manager_get_default_path ();
```

Get ladspa manager default plugin path.

Returns

the plugin default search path as a string vector.

[element-type utf8][array zero-terminated=1][transfer none]

Since: 3.0.0

ags_ladspa_manager_set_default_path ()

```
void
ags_ladspa_manager_set_default_path (gchar **default_path);
```

Set ladspa manager default plugin path.

Parameters

default_path	the string vector array to use as default path.	*[element-type utf8][array zero-terminated=1][transfer full]*

Since: 3.0.0

ags_ladspa_manager_get_filenames ()

```
gchar~**
ags_ladspa_manager_get_filenames (AgsLadspaManager *ladspa_manager);
```

Retrieve all filenames

Parameters

ladspa_manager	the AgsLadspaManager	

Returns

a NULL-terminated array of filenames.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ladspa_manager_find_ladspa_plugin ()

```
AgsLadspaPlugin~*
ags_ladspa_manager_find_ladspa_plugin (AgsLadspaManager *ladspa_manager,
                                       gchar *filename,
                                       gchar *effect);
```

Lookup filename in loaded plugins.

Parameters

ladspa_manager	the AgsLadspaManager	
filename	the filename of the plugin	
effect	the effect's name	

Returns

the AgsLadspaPlugin.

[transfer none]

Since: 3.0.0

ags_ladspa_manager_find_ladspa_plugin_with_fallback ()

```
AgsLadspaPlugin~*
ags_ladspa_manager_find_ladspa_plugin_with_fallback
                                (AgsLadspaManager *ladspa_manager,
                                 gchar *filename,
                                 gchar *effect);
```

Lookup filename in loaded plugins.

Parameters

ladspa_manager	the AgsLadspaManager	
filename	the filename of the plugin	
effect	the effect's name	

Returns

the AgsLadspaPlugin.

[transfer none]

Since: 3.5.10

ags_ladspa_manager_load_blacklist ()

```
void
ags_ladspa_manager_load_blacklist (AgsLadspaManager *ladspa_manager,
                                   gchar *blacklist_filename);
```

Load blacklisted plugin filenames.

Parameters

ladspa_manager	the AgsLadspaManager	
blacklist_filename	the filename as string	

Since: 3.0.0

ags_ladspa_manager_load_file ()

```
void
ags_ladspa_manager_load_file (AgsLadspaManager *ladspa_manager,
                              gchar *ladspa_path,
                              gchar *filename);
```

Load *filename* specified plugin.

Parameters

ladspa_manager	the AgsLadspaManager	
ladspa_path	the LADSPA path	
filename	the filename of the plugin	

Since: 3.0.0

ags_ladspa_manager_load_default_directory ()

```
void
ags_ladspa_manager_load_default_directory
                                (AgsLadspaManager *ladspa_manager);
```

Loads all available plugins.

Parameters

ladspa_manager	the AgsLadspaManager	

Since: 3.0.0

ags_ladspa_manager_get_ladspa_plugin ()

```
GList~*
ags_ladspa_manager_get_ladspa_plugin (AgsLadspaManager *ladspa_manager);
```

Get ladspa plugin.

Parameters

ladspa_manager	the AgsLadspaManager	

Returns

the GList containing AgsLadspaPlugin.

[transfer full]

Since: 3.5.15

ags_ladspa_manager_get_instance ()

```
AgsLadspaManager~*
ags_ladspa_manager_get_instance ();
```

Get instance.

Returns

the AgsLadspaManager.

[transfer none]

Since: 3.0.0

ags_ladspa_manager_new ()

```
AgsLadspaManager~*
ags_ladspa_manager_new ();
```

Create a new instance of AgsLadspaManager

Returns

the new AgsLadspaManager

Since: 3.0.0

AGS_IS_LADSPA_MANAGER()

```
#define AGS_IS_LADSPA_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LADSPA_MANAGER))
```

AGS_IS_LADSPA_MANAGER_CLASS()

```
#define AGS_IS_LADSPA_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LADSPA_MANAGER))
```

AGS_LADSPA_MANAGER()

```
#define AGS_LADSPA_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LADSPA_MANAGER, AgsLadspaManager))
```

AGS_LADSPA_MANAGER_CLASS()

```
#define AGS_LADSPA_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LADSPA_MANAGER, AgsLadspaManagerClass))
```

AGS_LADSPA_MANAGER_GET_CLASS()

```
#define AGS_LADSPA_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LADSPA_MANAGER, AgsLadspaManagerClass))
```

ags_ladspa_manager_get_type ()

```
GType
ags_ladspa_manager_get_type (void);
```

Types and Values

AGS_TYPE_LADSPA_MANAGER

```
#define AGS_TYPE_LADSPA_MANAGER                 (ags_ladspa_manager_get_type())
```

struct AgsLadspaManager

```
struct AgsLadspaManager;
```

struct AgsLadspaManagerClass

```
struct AgsLadspaManagerClass {
  GObjectClass gobject;
};
```

2.3 AgsLadspaPlugin

AgsLadspaPlugin — The ladspa plugin class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LADSPA_PLUGIN_DESCRIPTOR()
AgsLadspaPlugin *	ags_ladspa_plugin_new ()
#define	AGS_IS_LADSPA_PLUGIN()
#define	AGS_IS_LADSPA_PLUGIN_CLASS()
#define	AGS_LADSPA_PLUGIN()
#define	AGS_LADSPA_PLUGIN_CLASS()
#define	AGS_LADSPA_PLUGIN_GET_CLASS()
GType	ags_ladspa_plugin_get_type ()

Properties

guint	unique-id	Read / Write

Types and Values

#define	AGS_TYPE_LADSPA_PLUGIN
struct	AgsLadspaPlugin
struct	AgsLadspaPluginClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsBasePlugin
        &#x2570;&#x2500;&#x2500; AgsLadspaPlugin
```

Includes

```
#include <ags/plugin/ags_ladspa_plugin.h>
```

Description

The AgsLadspaPlugin loads/unloads a Ladspa plugin.

Functions

AGS_LADSPA_PLUGIN_DESCRIPTOR()

```
#define AGS_LADSPA_PLUGIN_DESCRIPTOR(ptr) ((LADSPA_Descriptor *)(ptr))
```

ags_ladspa_plugin_new ()

```
AgsLadspaPlugin~*
ags_ladspa_plugin_new (gchar *filename,
                       gchar *effect,
                       guint effect_index);
```

Creates an AgsLadspaPlugin

Parameters

filename	the plugin .so	
effect	the effect's string representation	
effect_index	the effect's index	

Returns

a new AgsLadspaPlugin

Since: 3.0.0

AGS_IS_LADSPA_PLUGIN()

```
#define AGS_IS_LADSPA_PLUGIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LADSPA_PLUGIN))
```

AGS_IS_LADSPA_PLUGIN_CLASS()

```
#define AGS_IS_LADSPA_PLUGIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LADSPA_PLUGIN))
```

AGS_LADSPA_PLUGIN()

```
#define AGS_LADSPA_PLUGIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LADSPA_PLUGIN, AgsLadspaPlugin))
```

AGS_LADSPA_PLUGIN_CLASS()

```
#define AGS_LADSPA_PLUGIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LADSPA_PLUGIN, AgsLadspaPluginClass))
```

AGS_LADSPA_PLUGIN_GET_CLASS()

```
#define AGS_LADSPA_PLUGIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LADSPA_PLUGIN, AgsLadspaPluginClass))
```

ags_ladspa_plugin_get_type ()

```
GType
ags_ladspa_plugin_get_type (void);
```

Types and Values

AGS_TYPE_LADSPA_PLUGIN

```
#define AGS_TYPE_LADSPA_PLUGIN                (ags_ladspa_plugin_get_type())
```

struct AgsLadspaPlugin

```
struct AgsLadspaPlugin;
```

struct AgsLadspaPluginClass

```
struct AgsLadspaPluginClass {
  AgsBasePluginClass base_plugin;
};
```

Property Details

The "unique-id" property

```
  "unique-id"                guint
```

The assigned unique-id.

Owner: AgsLadspaPlugin

Flags: Read / Write

Default value: 0

Since: 3.0.0

Chapter 3

DSSI plugins

The DSSI plugin objects.

3.1 AgsDssiPlugin

AgsDssiPlugin — The dssi plugin class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_DSSI_PLUGIN_DESCRIPTOR()
void	ags_dssi_plugin_change_program ()
AgsDssiPlugin *	ags_dssi_plugin_new ()
#define	AGS_DSSI_PLUGIN()
#define	AGS_DSSI_PLUGIN_CLASS()
#define	AGS_DSSI_PLUGIN_GET_CLASS()
#define	AGS_IS_DSSI_PLUGIN()
#define	AGS_IS_DSSI_PLUGIN_CLASS()
GType	ags_dssi_plugin_get_type ()

Properties

char *	program	Read / Write
guint	unique-id	Read / Write

Signals

void	change-program	Run Last

Types and Values

#define	AGS_TYPE_DSSI_PLUGIN
struct	AgsDssiPlugin
struct	AgsDssiPluginClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsBasePlugin
        &#x2570;&#x2500;&#x2500; AgsDssiPlugin
```

Includes

```
#include <ags/plugin/ags_dssi_plugin.h>
```

Description

The AgsDssiPlugin loads/unloads a Dssi plugin.

Functions

AGS_DSSI_PLUGIN_DESCRIPTOR()

```
#define AGS_DSSI_PLUGIN_DESCRIPTOR(ptr) ((DSSI_Descriptor *)(ptr))
```

ags_dssi_plugin_change_program ()

```
void
ags_dssi_plugin_change_program (AgsDssiPlugin *dssi_plugin,
                                gpointer ladspa_handle,
                                guint bank_index,
                                guint program_index);
```

ags_dssi_plugin_new ()

```
AgsDssiPlugin~*
ags_dssi_plugin_new (gchar *filename,
                     gchar *effect,
                     guint effect_index);
```

Create a new instance of AgsDssiPlugin

Parameters

filename	the plugin .so	
effect	the effect's string representation	
effect_index	the effect's index	

Returns

the new AgsDssiPlugin

Since: 3.0.0

AGS_DSSI_PLUGIN()

```
#define AGS_DSSI_PLUGIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_DSSI_PLUGIN, AgsDssiPlugin))
```

AGS_DSSI_PLUGIN_CLASS()

```
#define AGS_DSSI_PLUGIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_DSSI_PLUGIN, AgsDssiPluginClass))
```

AGS_DSSI_PLUGIN_GET_CLASS()

```
#define AGS_DSSI_PLUGIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_DSSI_PLUGIN, AgsDssiPluginClass))
```

AGS_IS_DSSI_PLUGIN()

```
#define AGS_IS_DSSI_PLUGIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_DSSI_PLUGIN))
```

AGS_IS_DSSI_PLUGIN_CLASS()

```
#define AGS_IS_DSSI_PLUGIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_DSSI_PLUGIN))
```

ags_dssi_plugin_get_type ()

```
GType
ags_dssi_plugin_get_type (void);
```

Types and Values

AGS_TYPE_DSSI_PLUGIN

```
#define AGS_TYPE_DSSI_PLUGIN                (ags_dssi_plugin_get_type())
```

struct AgsDssiPlugin

```
struct AgsDssiPlugin;
```

struct AgsDssiPluginClass

```
struct AgsDssiPluginClass {
  AgsBasePluginClass base_plugin;

  void (*change_program)(AgsDssiPlugin *dssi_plugin,
       gpointer ladspa_handle,
       guint bank_index,
       guint program_index);
};
```

Property Details

The "program" property

```
  "program"                  char~*
```

The program this plugin is located in.

Owner: AgsDssiPlugin

Flags: Read / Write

Default value: NULL

The "unique-id" property

```
  "unique-id"                guint
```

The assigned unique-id.

Owner: AgsDssiPlugin

Flags: Read / Write

Default value: 0

Since: 3.0.0

Signal Details

The "change-program" signal

```
void
user_function (AgsDssiPlugin *dssi_plugin,
               gpointer       ladspa_handle,
               guint          bank,
               guint          program,
               gpointer       user_data)
```

The ::change-program signal creates a new instance of plugin.

Parameters

dssi_plugin	the plugin to change-program	
ladspa_handle	the LADSPA handle	
bank	the bank number	
program	the program number	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

3.2 AgsDssiManager

AgsDssiManager — Singleton pattern to organize DSSI

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_DSSI_MANAGER_GET_OBJ_MUTEX()
gchar **	ags_dssi_manager_get_default_path ()
void	ags_dssi_manager_set_default_path ()
gchar **	ags_dssi_manager_get_filenames ()
AgsDssiPlugin *	ags_dssi_manager_find_dssi_plugin ()
AgsDssiPlugin *	ags_dssi_manager_find_dssi_plugin_with_fallback ()
void	ags_dssi_manager_load_blacklist ()
void	ags_dssi_manager_load_file ()
void	ags_dssi_manager_load_default_directory ()
GList *	ags_dssi_manager_get_dssi_plugin ()
AgsDssiManager *	ags_dssi_manager_get_instance ()
AgsDssiManager *	ags_dssi_manager_new ()
#define	AGS_DSSI_MANAGER()
#define	AGS_DSSI_MANAGER_CLASS()
#define	AGS_DSSI_MANAGER_GET_CLASS()
#define	AGS_IS_DSSI_MANAGER()
#define	AGS_IS_DSSI_MANAGER_CLASS()
GType	ags_dssi_manager_get_type ()

Types and Values

#define	AGS_TYPE_DSSI_MANAGER
struct	AgsDssiManager
struct	AgsDssiManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsDssiManager
```

Includes

```
#include <ags/plugin/ags_dssi_manager.h>
```

Description

The AgsDssiManager loads/unloads DSSI plugins.

Functions

AGS_DSSI_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_DSSI_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsDssiManager *) obj)->obj_mutex))
```

ags_dssi_manager_get_default_path ()

```
gchar~**
ags_dssi_manager_get_default_path ();
```

Get dssi manager default plugin path.

Returns

the plugin default search path as a string vector.

[element-type utf8][array zero-terminated=1][transfer none]

Since: 3.0.0

ags_dssi_manager_set_default_path ()

```
void
ags_dssi_manager_set_default_path (gchar **default_path);
```

Set dssi manager default plugin path.

Parameters

default_path	the string vector array to use as default path.	*[element-type utf8][array zero-terminated=1][transfer full]*

Since: 3.0.0

ags_dssi_manager_get_filenames ()

```
gchar~**
ags_dssi_manager_get_filenames (AgsDssiManager *dssi_manager);
```

Retrieve all filenames

Parameters

dssi_manager	the AgsDssiManager

Returns

a NULL-terminated array of filenames.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_dssi_manager_find_dssi_plugin ()

```
AgsDssiPlugin~*
ags_dssi_manager_find_dssi_plugin (AgsDssiManager *dssi_manager,
                                   gchar *filename,
                                   gchar *effect);
```

Lookup filename in loaded plugins.

Parameters

dssi_manager	the AgsDssiManager	
filename	the filename of the plugin	
effect	the effect's name	

Returns

the matching AgsDssiPlugin.

[transfer none]

Since: 3.0.0

ags_dssi_manager_find_dssi_plugin_with_fallback ()

```
AgsDssiPlugin~*
ags_dssi_manager_find_dssi_plugin_with_fallback
                                (AgsDssiManager *dssi_manager,
                                 gchar *filename,
                                 gchar *effect);
```

Lookup filename in loaded plugins.

Parameters

dssi_manager	the AgsDssiManager	
filename	the filename of the plugin	
effect	the effect's name	

Returns

the matching AgsDssiPlugin.

[transfer none]

Since: 3.5.10

ags_dssi_manager_load_blacklist ()

```
void
ags_dssi_manager_load_blacklist (AgsDssiManager *dssi_manager,
                                 gchar *blacklist_filename);
```

Load blacklisted plugin filenames.

Parameters

dssi_manager	the AgsDssiManager	
blacklist_filename	the filename as string	

Since: 3.0.0

ags_dssi_manager_load_file ()

```
void
ags_dssi_manager_load_file (AgsDssiManager *dssi_manager,
                            gchar *dssi_path,
                            gchar *filename);
```

Load *filename* specified plugin.

Parameters

dssi_manager	the AgsDssiManager	
dssi_path	the dssi path	
filename	the filename of the plugin	

Since: 3.0.0

ags_dssi_manager_load_default_directory ()

```
void
ags_dssi_manager_load_default_directory
                                (AgsDssiManager *dssi_manager);
```

Loads all available plugins.

Parameters

dssi_manager	the AgsDssiManager

Since: 3.0.0

ags_dssi_manager_get_dssi_plugin ()

```
GList~*
ags_dssi_manager_get_dssi_plugin (AgsDssiManager *dssi_manager);
```

Get dssi plugin.

Parameters

dssi_manager	the AgsDssiManager

Returns

the GList containing AgsDssiPlugin.

[transfer full]

Since: 3.5.15

ags_dssi_manager_get_instance ()

```
AgsDssiManager~*
ags_dssi_manager_get_instance ();
```

Get instance.

Returns

the AgsDssiManager.

[transfer none]

Since: 3.0.0

ags_dssi_manager_new ()

```
AgsDssiManager~*
ags_dssi_manager_new ();
```

Create a new instance of AgsDssiManager

Returns

the new AgsDssiManager

Since: 3.0.0

AGS_DSSI_MANAGER()

```
#define AGS_DSSI_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_DSSI_MANAGER, AgsDssiManager))
```

AGS_DSSI_MANAGER_CLASS()

```
#define AGS_DSSI_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_DSSI_MANAGER, AgsDssiManagerClass))
```

AGS_DSSI_MANAGER_GET_CLASS()

```
#define AGS_DSSI_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_DSSI_MANAGER, AgsDssiManagerClass))
```

AGS_IS_DSSI_MANAGER()

```
#define AGS_IS_DSSI_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_DSSI_MANAGER))
```

AGS_IS_DSSI_MANAGER_CLASS()

```
#define AGS_IS_DSSI_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_DSSI_MANAGER))
```

ags_dssi_manager_get_type ()

```
GType
ags_dssi_manager_get_type (void);
```

Types and Values

AGS_TYPE_DSSI_MANAGER

```
#define AGS_TYPE_DSSI_MANAGER                (ags_dssi_manager_get_type())
```

struct AgsDssiManager

```
struct AgsDssiManager;
```

struct AgsDssiManagerClass

```
struct AgsDssiManagerClass {
  GObjectClass gobject;
};
```

Chapter 4

LV2 plugins

The LV2 plugin objects.

4.1 AgsLv2Conversion

AgsLv2Conversion — Conversion of values

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_lv2_conversion_test_flags ()
void	ags_lv2_conversion_set_flags ()
void	ags_lv2_conversion_unset_flags ()
AgsLv2Conversion *	ags_lv2_conversion_new ()
#define	AGS_IS_LV2_CONVERSION()
#define	AGS_IS_LV2_CONVERSION_CLASS()
#define	AGS_LV2_CONVERSION()
#define	AGS_LV2_CONVERSION_CLASS()
#define	AGS_LV2_CONVERSION_GET_CLASS()
GType	ags_lv2_conversion_flags_get_type ()
GType	ags_lv2_conversion_get_type ()

Properties

double	lower	Read / Write
double	step-count	Read / Write
double	upper	Read / Write

Types and Values

#define	AGS_LV2_CONVERSION_DEFAULT_LOWER

#define	AGS_LV2_CONVERSION_DEFAULT_UPPER
#define	AGS_LV2_CONVERSION_DEFAULT_STEP_COUNT
enum	AgsLv2ConversionFlags
#define	AGS_TYPE_LV2_CONVERSION
struct	AgsLv2Conversion
struct	AgsLv2ConversionClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsLv2ConversionFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsConversion
        &#x2570;&#x2500;&#x2500; AgsLv2Conversion
```

Includes

```
#include <ags/plugin/ags_lv2_conversion.h>
```

Description

The AgsLv2Conversion converts values.

Functions

ags_lv2_conversion_test_flags ()

```
gboolean
ags_lv2_conversion_test_flags (AgsLv2Conversion *lv2_conversion,
                               guint flags);
```

Test *flags* to be set on *recall* .

Parameters

lv2_conversion	the AgsLv2Conversion	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_lv2_conversion_set_flags ()

```
void
ags_lv2_conversion_set_flags (AgsLv2Conversion *lv2_conversion,
                              guint flags);
```

Set flags.

Parameters

lv2_conversion	the AgsLv2Conversion	
flags	the flags	

Since: 3.0.0

ags_lv2_conversion_unset_flags ()

```
void
ags_lv2_conversion_unset_flags (AgsLv2Conversion *lv2_conversion,
                                guint flags);
```

Unset flags.

Parameters

lv2_conversion	the AgsLv2Conversion	
flags	the flags	

Since: 3.0.0

ags_lv2_conversion_new ()

```
AgsLv2Conversion~*
ags_lv2_conversion_new ();
```

Create a new instance of AgsLv2Conversion.

Returns

the new AgsLv2Conversion

Since: 3.0.0

AGS_IS_LV2_CONVERSION()

```
#define AGS_IS_LV2_CONVERSION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CONVERSION))
```

AGS_IS_LV2_CONVERSION_CLASS()

```
#define AGS_IS_LV2_CONVERSION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CONVERSION))
```

AGS_LV2_CONVERSION()

```
#define AGS_LV2_CONVERSION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CONVERSION, AgsLv2Conversion))
```

AGS_LV2_CONVERSION_CLASS()

```
#define AGS_LV2_CONVERSION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CONVERSION, AgsLv2ConversionClass))
```

AGS_LV2_CONVERSION_GET_CLASS()

```
#define AGS_LV2_CONVERSION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_CONVERSION, AgsLv2ConversionClass))
```

ags_lv2_conversion_flags_get_type ()

```
GType
ags_lv2_conversion_flags_get_type ();
```

ags_lv2_conversion_get_type ()

```
GType
ags_lv2_conversion_get_type (void);
```

Types and Values

AGS_LV2_CONVERSION_DEFAULT_LOWER

```
#define AGS_LV2_CONVERSION_DEFAULT_LOWER (0.00001)
```

AGS_LV2_CONVERSION_DEFAULT_UPPER

```
#define AGS_LV2_CONVERSION_DEFAULT_UPPER (1.0)
```

AGS_LV2_CONVERSION_DEFAULT_STEP_COUNT

```
#define AGS_LV2_CONVERSION_DEFAULT_STEP_COUNT (1025.0)
```

enum AgsLv2ConversionFlags

Enum values to control the behavior or indicate internal state of AgsLv2Conversion by enable/disable as flags.

Members

AGS_LV2_CONVERSION_LOGARITHMIC	logarithmic

AGS_TYPE_LV2_CONVERSION

```
#define AGS_TYPE_LV2_CONVERSION                (ags_lv2_conversion_get_type())
```

struct AgsLv2Conversion

```
struct AgsLv2Conversion;
```

struct AgsLv2ConversionClass

```
struct AgsLv2ConversionClass {
  AgsConversionClass conversion;
};
```

Property Details

The "lower" property

```
  "lower"                    double
```

The lower to be used.

Owner: AgsLv2Conversion

Flags: Read / Write

Default value: 1e-05

Since: 3.0.0

The "step-count" property

```
  "step-count"               double
```

The step count to be used.

Owner: AgsLv2Conversion

Flags: Read / Write

Allowed values: >= 0

Default value: 1025

Since: 3.0.0

The "upper" property

```
  "upper"                    double
```

The upper to be used.

Owner: AgsLv2Conversion

Flags: Read / Write

Default value: 1

Since: 3.0.0

4.2 AgsLv2EventManager

AgsLv2EventManager — event manager

Stability Level

Stable, unless otherwise indicated

Functions

uint32_t	ags_lv2_event_manager_lv2_event_ref ()
uint32_t	ags_lv2_event_manager_lv2_event_unref ()
AgsLv2EventManager *	ags_lv2_event_manager_get_instance ()
AgsLv2EventManager *	ags_lv2_event_manager_new ()
#define	AGS_IS_LV2_EVENT_MANAGER()
#define	AGS_IS_LV2_EVENT_MANAGER_CLASS()
#define	AGS_LV2_EVENT_MANAGER()
#define	AGS_LV2_EVENT_MANAGER_CLASS()
#define	AGS_LV2_EVENT_MANAGER_GET_CLASS()
GType	ags_lv2_event_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2_EVENT_MANAGER
struct	AgsLv2EventManager
struct	AgsLv2EventManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsLv2EventManager
```

Includes

```
#include <ags/plugin/ags_lv2_event_manager.h>
```

Description

The AgsLv2EventManager allows you to do memory management with LV2_Event's.

Functions

ags_lv2_event_manager_lv2_event_ref ()

```
uint32_t
ags_lv2_event_manager_lv2_event_ref (LV2_Event_Callback_Data callback_data,
                                     LV2_Event *event);
```

ags_lv2_event_manager_lv2_event_unref ()

```
uint32_t
ags_lv2_event_manager_lv2_event_unref (LV2_Event_Callback_Data callback_data,
                                       LV2_Event *event);
```

ags_lv2_event_manager_get_instance ()

```
AgsLv2EventManager~*
ags_lv2_event_manager_get_instance ();
```

Singleton function to optain the id manager instance.

Returns

an instance of AgsLv2EventManager.

[transfer none]

Since: 3.0.0

ags_lv2_event_manager_new ()

```
AgsLv2EventManager~*
ags_lv2_event_manager_new ();
```

Instantiate a id manager.

Returns

a new AgsLv2EventManager

Since: 3.0.0

AGS_IS_LV2_EVENT_MANAGER()

```
#define AGS_IS_LV2_EVENT_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_EVENT_MANAGER))
```

AGS_IS_LV2_EVENT_MANAGER_CLASS()

```
#define AGS_IS_LV2_EVENT_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_EVENT_MANAGER))
```

AGS_LV2_EVENT_MANAGER()

```
#define AGS_LV2_EVENT_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_EVENT_MANAGER, AgsLv2EventManager))
```

AGS_LV2_EVENT_MANAGER_CLASS()

```
#define AGS_LV2_EVENT_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_EVENT_MANAGER, AgsLv2EventManagerClass))
```

AGS_LV2_EVENT_MANAGER_GET_CLASS()

```
#define AGS_LV2_EVENT_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_EVENT_MANAGER, AgsLv2EventManagerClass))
```

ags_lv2_event_manager_get_type ()

```
GType
ags_lv2_event_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2_EVENT_MANAGER

```
#define AGS_TYPE_LV2_EVENT_MANAGER                (ags_lv2_event_manager_get_type())
```

struct AgsLv2EventManager

```
struct AgsLv2EventManager;
```

struct AgsLv2EventManagerClass

```
struct AgsLv2EventManagerClass {
  GObjectClass gobject;
};
```

4.3 AgsLv2LogManager

AgsLv2LogManager — allow lv2 plugins to log

Stability Level

Stable, unless otherwise indicated

Functions

int	ags_lv2_log_manager_printf ()
int	ags_lv2_log_manager_vprintf ()
AgsLv2LogManager *	ags_lv2_log_manager_get_instance ()
AgsLv2LogManager *	ags_lv2_log_manager_new ()
#define	AGS_IS_LV2_LOG_MANAGER()
#define	AGS_IS_LV2_LOG_MANAGER_CLASS()
#define	AGS_LV2_LOG_MANAGER()
#define	AGS_LV2_LOG_MANAGER_CLASS()
#define	AGS_LV2_LOG_MANAGER_GET_CLASS()
GType	ags_lv2_log_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2_LOG_MANAGER
struct	AgsLv2LogManager
struct	AgsLv2LogManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsLv2LogManager
```

Includes

```
#include <ags/plugin/ags_lv2_log_manager.h>
```

Description

The AgsLv2LogManager handles console IO.

Functions

ags_lv2_log_manager_printf ()

```
int
ags_lv2_log_manager_printf (LV2_Log_Handle handle,
                            LV2_URID type,
                            const char *fmt,
                            ...);
```

Print formatted.

Parameters

handle	the log handle.	*[type gpointer][transfer none]*
type	LV2 URID.	*[type gpointer][transfer none]*
fmt	format	
...	NULL terminated variable arguments	

Returns

0

Since: 3.0.0

ags_lv2_log_manager_vprintf ()

```
int
ags_lv2_log_manager_vprintf (LV2_Log_Handle handle,
                             LV2_URID type,
                             const char *fmt,
                             va_list ap);
```

Variadict print formatted.

Parameters

handle	the log handle.	*[type gpointer][transfer none]*
type	LV2 URID.	*[type gpointer][transfer none]*
fmt	format	
ap	argument pointer	

Returns

0

Since: 3.0.0

ags_lv2_log_manager_get_instance ()

```
AgsLv2LogManager~*
ags_lv2_log_manager_get_instance ();
```

Singleton function to optain the id manager instance.

Returns

an instance of AgsLv2LogManager.

[transfer none]

Since: 3.0.0

ags_lv2_log_manager_new ()

```
AgsLv2LogManager~*
ags_lv2_log_manager_new ();
```

Instantiate a id manager.

Returns

a new AgsLv2LogManager

Since: 3.0.0

AGS_IS_LV2_LOG_MANAGER()

```
#define AGS_IS_LV2_LOG_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_LOG_MANAGER))
```

AGS_IS_LV2_LOG_MANAGER_CLASS()

```
#define AGS_IS_LV2_LOG_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_LOG_MANAGER))
```

AGS_LV2_LOG_MANAGER()

```
#define AGS_LV2_LOG_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_LOG_MANAGER, AgsLv2LogManager))
```

AGS_LV2_LOG_MANAGER_CLASS()

```
#define AGS_LV2_LOG_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_LOG_MANAGER, AgsLv2LogManagerClass))
```

AGS_LV2_LOG_MANAGER_GET_CLASS()

```
#define AGS_LV2_LOG_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_LOG_MANAGER, AgsLv2LogManagerClass))
```

ags_lv2_log_manager_get_type ()

```
GType
ags_lv2_log_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2_LOG_MANAGER

```
#define AGS_TYPE_LV2_LOG_MANAGER                (ags_lv2_log_manager_get_type())
```

struct AgsLv2LogManager

```
struct AgsLv2LogManager;
```

struct AgsLv2LogManagerClass

```
struct AgsLv2LogManagerClass {
  GObjectClass gobject;
};
```

4.4 AgsLv2Manager

AgsLv2Manager — Singleton pattern to organize LV2

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_MANAGER_GET_OBJ_MUTEX()
gboolean	ags_lv2_manager_global_get_parse_names ()
gboolean	ags_lv2_manager_global_get_preserve_turtle ()
gchar **	ags_lv2_manager_get_default_path ()
void	ags_lv2_manager_set_default_path ()
gchar **	ags_lv2_manager_get_filenames ()
AgsLv2Plugin *	ags_lv2_manager_find_lv2_plugin ()
AgsLv2Plugin *	ags_lv2_manager_find_lv2_plugin_with_fallback ()
void	ags_lv2_manager_load_blacklist ()
void	ags_lv2_manager_load_file ()
void	ags_lv2_manager_load_preset ()
void	ags_lv2_manager_quick_scan_default_directory ()
void	ags_lv2_manager_load_default_directory ()
GList *	ags_lv2_manager_get_lv2_plugin ()
AgsLv2Manager *	ags_lv2_manager_get_instance ()
AgsLv2Manager *	ags_lv2_manager_new ()
#define	AGS_IS_LV2_MANAGER()
#define	AGS_IS_LV2_MANAGER_CLASS()
#define	AGS_LV2_MANAGER()
#define	AGS_LV2_MANAGER_CLASS()
#define	AGS_LV2_MANAGER_GET_CLASS()
GType	ags_lv2_manager_get_type ()

Properties

char *	locale	Read / Write

Types and Values

#define	AGS_LV2_MANAGER_DEFAULT_LOCALE
#define	AGS_TYPE_LV2_MANAGER
struct	AgsLv2Manager
struct	AgsLv2ManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLv2Manager
```

Includes

```
#include <ags/plugin/ags_lv2_manager.h>
```

Description

The AgsLv2Manager loads/unloads LV2 plugins.

Functions

AGS_LV2_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLv2Manager *) obj)->obj_mutex))
```

ags_lv2_manager_global_get_parse_names ()

```
gboolean
ags_lv2_manager_global_get_parse_names
                                ();
```

Get global config value parse names.

Returns

if TRUE parse only names, else not

Since: 3.0.0

ags_lv2_manager_global_get_preserve_turtle ()

```
gboolean
ags_lv2_manager_global_get_preserve_turtle
                                ();
```

Get global config value preserve turtle.

Returns

if TRUE preserve turtles, else not

Since: 3.0.0

ags_lv2_manager_get_default_path ()

```
gchar~**
ags_lv2_manager_get_default_path ();
```

Get lv2 manager default plugin path.

Returns

the plugin default search path as a string vector.

[element-type utf8][array zero-terminated=1][transfer none]

Since: 3.0.0

ags_lv2_manager_set_default_path ()

```
void
ags_lv2_manager_set_default_path (gchar **default_path);
```

Set lv2 manager default plugin path.

Parameters

default_path	the string vector array to use as default path.	*[element-type utf8][array zero-terminated=1][transfer full]*

Since: 3.0.0

ags_lv2_manager_get_filenames ()

```
gchar~**
ags_lv2_manager_get_filenames (AgsLv2Manager *lv2_manager);
```

Retrieve all filenames

Parameters

lv2_manager	the AgsLv2Manager	

Returns

a NULL-terminated array of filenames.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_lv2_manager_find_lv2_plugin ()

```
AgsLv2Plugin~*
ags_lv2_manager_find_lv2_plugin (AgsLv2Manager *lv2_manager,
                                 gchar *filename,
                                 gchar *effect);
```

Lookup filename in loaded plugins.

Parameters

lv2_manager	the AgsLv2Manager	
filename	the filename of the plugin	
effect	the effect's name	

Returns

the AgsLv2Plugin.

[transfer none]

Since: 3.0.0

ags_lv2_manager_find_lv2_plugin_with_fallback ()

```
AgsLv2Plugin~*
ags_lv2_manager_find_lv2_plugin_with_fallback
                                (AgsLv2Manager *lv2_manager,
                                 gchar *filename,
                                 gchar *effect);
```

Lookup filename in loaded plugins.

Parameters

lv2_manager	the AgsLv2Manager	
filename	the filename of the plugin	
effect	the effect's name	

Returns

the AgsLv2Plugin.

[transfer none]

Since: 3.5.10

ags_lv2_manager_load_blacklist ()

```
void
ags_lv2_manager_load_blacklist (AgsLv2Manager *lv2_manager,
                                gchar *blacklist_filename);
```

Load blacklisted plugin filenames.

Parameters

lv2_manager	the AgsLv2Manager	
blacklist_filename	the filename as string	

Since: 3.0.0

ags_lv2_manager_load_file ()

```
void
ags_lv2_manager_load_file (AgsLv2Manager *lv2_manager,
                           AgsTurtle *manifest,
                           AgsTurtle *turtle,
                           gchar *lv2_path,
                           gchar *filename);
```

Load *filename* specified plugin.

Parameters

lv2_manager	the AgsLv2Manager	
manifest	the manifest	
turtle	the loaded turtle	
lv2_path	the lv2 path	
filename	the filename of the plugin	

Since: 3.0.0

ags_lv2_manager_load_preset ()

```
void
ags_lv2_manager_load_preset (AgsLv2Manager *lv2_manager,
                             AgsLv2Plugin *lv2_plugin,
                             AgsTurtle *preset);
```

Load preset.

Parameters

lv2_manager	the AgsLv2Manager	
lv2_plugin	the AgsLv2Plugin	
preset	the AgsTurtle	

Since: 3.0.0

ags_lv2_manager_quick_scan_default_directory ()

```
void
ags_lv2_manager_quick_scan_default_directory
                                (AgsLv2Manager *lv2_manager);
```

Quick scan available plugins.

Parameters

lv2_manager	the AgsLv2Manager	

Since: 3.2.7

ags_lv2_manager_load_default_directory ()

```
void
ags_lv2_manager_load_default_directory
                                (AgsLv2Manager *lv2_manager);
```

Loads all available plugins.

Parameters

lv2_manager	the AgsLv2Manager

Since: 3.0.0

ags_lv2_manager_get_lv2_plugin ()

```
GList~*
ags_lv2_manager_get_lv2_plugin (AgsLv2Manager *lv2_manager);
```

Get lv2 plugin.

Parameters

lv2_manager	the AgsLv2Manager

Returns

the GList containing AgsLv2Plugin.

[transfer full]

Since: 3.5.15

ags_lv2_manager_get_instance ()

```
AgsLv2Manager~*
ags_lv2_manager_get_instance ();
```

Get instance.

Returns

the AgsLv2Manager.

[transfer none]

Since: 3.0.0

ags_lv2_manager_new ()

```
AgsLv2Manager~*
ags_lv2_manager_new (gchar *locale);
```

Create a new instance of AgsLv2Manager

Parameters

locale	the default locale

Returns

the new AgsLv2Manager

Since: 3.0.0

AGS_IS_LV2_MANAGER()

```
#define AGS_IS_LV2_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_MANAGER))
```

AGS_IS_LV2_MANAGER_CLASS()

```
#define AGS_IS_LV2_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_MANAGER))
```

AGS_LV2_MANAGER()

```
#define AGS_LV2_MANAGER(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_MANAGER, AgsLv2Manager))
```

AGS_LV2_MANAGER_CLASS()

```
#define AGS_LV2_MANAGER_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_MANAGER, AgsLv2ManagerClass))
```

AGS_LV2_MANAGER_GET_CLASS()

```
#define AGS_LV2_MANAGER_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_MANAGER, AgsLv2ManagerClass))
```

ags_lv2_manager_get_type ()

```
GType
ags_lv2_manager_get_type (void);
```

Types and Values

AGS_LV2_MANAGER_DEFAULT_LOCALE

```
#define AGS_LV2_MANAGER_DEFAULT_LOCALE "en-gb"
```

AGS_TYPE_LV2_MANAGER

```
#define AGS_TYPE_LV2_MANAGER                   (ags_lv2_manager_get_type())
```

struct AgsLv2Manager

```
struct AgsLv2Manager;
```

struct AgsLv2ManagerClass

```
struct AgsLv2ManagerClass {
  GObjectClass gobject;
};
```

Property Details

The "locale" property

```
  "locale"                    char~*
```

The assigned locale.

Owner: AgsLv2Manager

Flags: Read / Write

Default value: NULL

Since: 3.0.0

4.5 AgsLv2OptionManager

AgsLv2OptionManager — option manager

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_OPTION_MANAGER_GET_OBJ_MUTEX()
#define	AGS_LV2_OPTION_RESSOURCE()
#define	AGS_LV2_OPTIONS_OPTION()
AgsLv2OptionRessource *	ags_lv2_option_ressource_alloc ()
gboolean	ags_lv2_option_manager_ressource_insert ()
gboolean	ags_lv2_option_manager_ressource_remove ()
gpointer	ags_lv2_option_manager_ressource_lookup ()
gboolean	ags_lv2_option_manager_ressource_lookup_extended ()
void	ags_lv2_option_manager_get_option ()
void	ags_lv2_option_manager_set_option ()
uint32_t	ags_lv2_option_manager_lv2_options_get ()
uint32_t	ags_lv2_option_manager_lv2_options_set ()
AgsLv2OptionManager *	ags_lv2_option_manager_get_instance ()
AgsLv2OptionManager *	ags_lv2_option_manager_new ()
#define	AGS_IS_LV2_OPTION_MANAGER()
#define	AGS_IS_LV2_OPTION_MANAGER_CLASS()
#define	AGS_LV2_OPTION_MANAGER()
#define	AGS_LV2_OPTION_MANAGER_CLASS()
#define	AGS_LV2_OPTION_MANAGER_GET_CLASS()
GType	ags_lv2_option_manager_get_type ()

Signals

void	get-option	Run Last
void	set-option	Run Last

Types and Values

struct	AgsLv2OptionRessource
#define	AGS_TYPE_LV2_OPTION_MANAGER
struct	AgsLv2OptionManager
struct	AgsLv2OptionManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsLv2OptionManager
```

Includes

```
#include <ags/plugin/ags_lv2_option_manager.h>
```

Description

The AgsLv2OptionManager gives you access to plugin instances global configuration. And stores instance related ressources.

Functions

AGS_LV2_OPTION_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2_OPTION_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLv2OptionManager *) obj)-> ↩
    obj_mutex))
```

AGS_LV2_OPTION_RESSOURCE()

```
#define AGS_LV2_OPTION_RESSOURCE(ptr) ((AgsLv2OptionRessource *)(ptr))
```

AGS_LV2_OPTIONS_OPTION()

```
#define AGS_LV2_OPTIONS_OPTION(ptr) ((LV2_Options_Option *)(ptr))
```

ags_lv2_option_ressource_alloc ()

```
AgsLv2OptionRessource~*
ags_lv2_option_ressource_alloc ();
```

Allocate an AgsLv2OptionRessource.

Returns

the newly created AgsLv2OptionRessource.

[type gpointer][transfer none]

Since: 3.0.0

ags_lv2_option_manager_ressource_insert ()

```
gboolean
ags_lv2_option_manager_ressource_insert
                                (AgsLv2OptionManager *lv2_option_manager,
                                 AgsLv2OptionRessource *lv2_option_ressource,
                                 gpointer data);
```

Inserts a data into hash associated with `lv2_option_ressource`.

Parameters

lv2_option_manager	the AgsLv2OptionManager	
lv2_option_ressource	the AgsLv2OptionRessource as key.	*[type gpointer][transfer none]*
data	the data	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_lv2_option_manager_ressource_remove ()

```
gboolean
ags_lv2_option_manager_ressource_remove
                                (AgsLv2OptionManager *lv2_option_manager,
                                 AgsLv2OptionRessource *lv2_option_ressource);
```

Removes an entry associated with `lv2_option_ressource`.

Parameters

lv2_option_manager	the AgsLv2OptionManager	
lv2_option_ressource	the struct to remove.	*[type gpointer][transfer none]*

Returns

TRUE as successfully removed, otherwise FALSE

Since: 3.0.0

ags_lv2_option_manager_ressource_lookup ()

```
gpointer
ags_lv2_option_manager_ressource_lookup
                                (AgsLv2OptionManager *lv2_option_manager,
                                 AgsLv2OptionRessource *lv2_option_ressource);
```

Lookup a ressource associated with *lv2_option_ressource* in *lv2_option_manager* .

Parameters

lv2_option_manager	the AgsLv2OptionManager	
lv2_option_ressource	the AgsLv2OptionRessource to lookup.	*[type gpointer][transfer none]*

Returns

the pointer on success, else NULL

Since: 3.0.0

ags_lv2_option_manager_ressource_lookup_extended ()

```
gboolean
ags_lv2_option_manager_ressource_lookup_extended
                                (AgsLv2OptionManager *lv2_option_manager,
                                 AgsLv2OptionRessource *lv2_option_ressource,
                                 gpointer *orig_key,
                                 gpointer *value);
```

Lookup a ressource associated with *lv2_option_ressource* in *lv2_option_manager* .

Parameters

lv2_option_manager	the AgsLv2OptionManager	
lv2_option_ressource	the AgsLv2OptionRessource to lookup	
orig_key	the original key found	
value	the matched value	

Returns

TRUE if ressource found, else FALSE

Since: 3.0.0

ags_lv2_option_manager_get_option ()

```
void
ags_lv2_option_manager_get_option (AgsLv2OptionManager *lv2_option_manager,
                                   gpointer instance,
                                   gpointer option,
                                   gpointer retval);
```

Get option.

Parameters

lv2_option_manager	the AgsLv2OptionManager	
instance	the instance	
option	the option	
retval	return value for LV2_Options_Status	

Since: 3.0.0

ags_lv2_option_manager_set_option ()

```
void
ags_lv2_option_manager_set_option (AgsLv2OptionManager *lv2_option_manager,
                                   gpointer instance,
                                   gpointer option,
                                   gpointer retval);
```

Set option.

Parameters

lv2_option_manager	the AgsLv2OptionManager	
instance	the instance	
option	the option	
retval	return value for LV2_Options_Status	

Since: 3.0.0

ags_lv2_option_manager_lv2_options_get ()

```
uint32_t
ags_lv2_option_manager_lv2_options_get
                                (LV2_Handle instance,
                                 LV2_Options_Option *options);
```

The LV2 options interface's get method.

Parameters

instance	the lv2 instance	
options	the LV2_Options	

Since: 3.0.0

ags_lv2_option_manager_lv2_options_set ()

```
uint32_t
ags_lv2_option_manager_lv2_options_set
                                (LV2_Handle instance,
                                 LV2_Options_Option *options);
```

The LV2 options interface's set method.

Parameters

instance	the lv2 instance	
options	the LV2_Options	

Since: 3.0.0

ags_lv2_option_manager_get_instance ()

```
AgsLv2OptionManager~*
ags_lv2_option_manager_get_instance ();
```

Singleton function to optain the id manager instance.

Returns

an instance of AgsLv2OptionManager.

[transfer none]

Since: 3.0.0

ags_lv2_option_manager_new ()

```
AgsLv2OptionManager~*
ags_lv2_option_manager_new ();
```

Instantiate a id manager.

Returns

a new AgsLv2OptionManager

Since: 3.0.0

AGS_IS_LV2_OPTION_MANAGER()

```
#define AGS_IS_LV2_OPTION_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_OPTION_MANAGER))
```

AGS_IS_LV2_OPTION_MANAGER_CLASS()

```
#define AGS_IS_LV2_OPTION_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_OPTION_MANAGER))
```

AGS_LV2_OPTION_MANAGER()

```
#define AGS_LV2_OPTION_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_OPTION_MANAGER, AgsLv2OptionManager))
```

AGS_LV2_OPTION_MANAGER_CLASS()

```
#define AGS_LV2_OPTION_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_OPTION_MANAGER, AgsLv2OptionManagerClass))
```

AGS_LV2_OPTION_MANAGER_GET_CLASS()

```
#define AGS_LV2_OPTION_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_OPTION_MANAGER, AgsLv2OptionManagerClass))
```

ags_lv2_option_manager_get_type ()

```
GType
ags_lv2_option_manager_get_type (void);
```

Types and Values

struct AgsLv2OptionRessource

```
struct AgsLv2OptionRessource {
  LV2_Handle instance;
  LV2_Options_Option *option;
};
```

AGS_TYPE_LV2_OPTION_MANAGER

```
#define AGS_TYPE_LV2_OPTION_MANAGER                (ags_lv2_option_manager_get_type())
```

struct AgsLv2OptionManager

```
struct AgsLv2OptionManager;
```

struct AgsLv2OptionManagerClass

```
struct AgsLv2OptionManagerClass {
  GObjectClass gobject;

  void (*get_option)(AgsLv2OptionManager *lv2_option_manager,
         gpointer instance,
         gpointer option,
         gpointer retval);
  void (*set_option)(AgsLv2OptionManager *lv2_option_manager,
         gpointer instance,
         gpointer option,
         gpointer retval);
};
```

Signal Details

The "get-option" signal

```
void
user_function (AgsLv2OptionManager *lv2_option_manager,
               gpointer             intstance,
               gpointer             option,
               gpointer             retval,
               gpointer             user_data)
```

The ::get-option signal gets options of the manager.

Parameters

lv2_option_manager	the plugin to instantiate	
intstance	the LV2 plugin instance	
option	the LV2 option pointer	
retval	the location of the return value	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "set-option" signal

```
void
user_function (AgsLv2OptionManager *lv2_option_manager,
               gpointer             intstance,
               gpointer             option,
               gpointer             retval,
               gpointer             user_data)
```

The ::set-option signal sets options for the manager.

Parameters

lv2_option_manager	the plugin to instantiate	
intstance	the LV2 plugin instance	
option	the LV2 option pointer	
retval	the location of the return value	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

4.6 AgsLv2TurtleParser

AgsLv2TurtleParser — The lv2 turtle parser class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_TURTLE_PARSER_GET_OBJ_MUTEX()
void	ags_lv2_turtle_parser_parse_names ()
void	ags_lv2_turtle_parser_parse ()
AgsLv2TurtleParser *	ags_lv2_turtle_parser_new ()
#define	AGS_IS_LV2_TURTLE_PARSER()
#define	AGS_IS_LV2_TURTLE_PARSER_CLASS()
#define	AGS_LV2_TURTLE_PARSER()
#define	AGS_LV2_TURTLE_PARSER_CLASS()
#define	AGS_LV2_TURTLE_PARSER_GET_CLASS()
GType	ags_lv2_turtle_parser_get_type ()

Properties

gpointer	plugin	Read / Write
gpointer	preset	Read / Write
gpointer	turtle	Read / Write
gpointer	ui-plugin	Read / Write

Types and Values

#define	AGS_TYPE_LV2_TURTLE_PARSER
struct	AgsLv2TurtleParser
struct	AgsLv2TurtleParserClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLv2TurtleParser
```

Includes

```
#include <ags/plugin/ags_lv2_turtle_parser.h>
```

Description

The AgsLv2TurtleParser parses RDF Turtle files.

Functions

AGS_LV2_TURTLE_PARSER_GET_OBJ_MUTEX()

```
#define AGS_LV2_TURTLE_PARSER_GET_OBJ_MUTEX(obj) (&(((AgsLv2TurtleParser *) obj)->obj_mutex ↩
    ))
```

ags_lv2_turtle_parser_parse_names ()

```
void
ags_lv2_turtle_parser_parse_names (AgsLv2TurtleParser *lv2_turtle_parser,
                                   AgsTurtle **turtle,
                                   guint n_turtle);
```

Parse names only from manifest and referred turtles.

Parameters

lv2_turtle_parser	the AgsLv2TurtleParser	
turtle	the NULL terminated array of AgsTurtle	
n_turtle	the turtle count	

Since: 3.0.0

ags_lv2_turtle_parser_parse ()

```
void
ags_lv2_turtle_parser_parse (AgsLv2TurtleParser *lv2_turtle_parser,
                             AgsTurtle **turtle,
                             guint n_turtle);
```

Parse manifest and referred turtles.

Parameters

lv2_turtle_parser	the AgsLv2TurtleParser	
turtle	the NULL terminated array of AgsTurtle.	*[element-type Ags.Turtle][array zero-terminated=1][inout]*
n_turtle	the turtle count	

Since: 3.0.0

ags_lv2_turtle_parser_new ()

```
AgsLv2TurtleParser~*
ags_lv2_turtle_parser_new (AgsTurtle *manifest);
```

Creates an AgsLv2TurtleParser

Parameters

manifest	the manifest as AgsTurtle

Returns

a new AgsLv2TurtleParser

Since: 3.0.0

AGS_IS_LV2_TURTLE_PARSER()

```
#define AGS_IS_LV2_TURTLE_PARSER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_TURTLE_PARSER))
```

AGS_IS_LV2_TURTLE_PARSER_CLASS()

```
#define AGS_IS_LV2_TURTLE_PARSER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_TURTLE_PARSER))
```

AGS_LV2_TURTLE_PARSER()

```
#define AGS_LV2_TURTLE_PARSER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_TURTLE_PARSER, AgsLv2TurtleParser))
```

AGS_LV2_TURTLE_PARSER_CLASS()

```
#define AGS_LV2_TURTLE_PARSER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_TURTLE_PARSER, AgsLv2TurtleParserClass))
```

AGS_LV2_TURTLE_PARSER_GET_CLASS()

```
#define AGS_LV2_TURTLE_PARSER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_TURTLE_PARSER, AgsLv2TurtleParserClass))
```

ags_lv2_turtle_parser_get_type ()

```
GType
ags_lv2_turtle_parser_get_type (void);
```

Types and Values

AGS_TYPE_LV2_TURTLE_PARSER

```
#define AGS_TYPE_LV2_TURTLE_PARSER                 (ags_lv2_turtle_parser_get_type())
```

struct AgsLv2TurtleParser

```
struct AgsLv2TurtleParser;
```

struct AgsLv2TurtleParserClass

```
struct AgsLv2TurtleParserClass {
  GObjectClass gobject;
};
```

Property Details

The "plugin" property

```
  "plugin"                    gpointer
```

The assigned GList containing AgsLv2Plugin.

[transfer full]

Owner: AgsLv2TurtleParser

Flags: Read / Write

Since: 3.0.0

The "preset" property

```
  "preset"                    gpointer
```

The assigned GList containing AgsLv2Preset.

[transfer full]

Owner: AgsLv2TurtleParser

Flags: Read / Write

Since: 3.0.0

The "turtle" property

```
  "turtle"                    gpointer
```

The assigned GList containing AgsTurtle.

[transfer full]

Owner: AgsLv2TurtleParser

Flags: Read / Write

Since: 3.0.0

The "ui-plugin" property

```
"ui-plugin"                 gpointer
```

The parsed lv2 UI plugins.

Owner: AgsLv2TurtleParser

Flags: Read / Write

4.7 AgsLv2TurtleScanner

AgsLv2TurtleScanner — The lv2 turtle scanner class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_CACHE_TURTLE()
#define	AGS_LV2_TURTLE_SCANNER_GET_OBJ_MUTEX()
AgsLv2CacheTurtle *	ags_lv2_cache_turtle_alloc ()
void	ags_lv2_cache_turtle_free ()
GList *	ags_lv2_cache_turtle_find ()
void	ags_lv2_turtle_scanner_quick_scan_see_also ()
void	ags_lv2_turtle_scanner_quick_scan ()
AgsLv2TurtleScanner *	ags_lv2_turtle_scanner_new ()
#define	AGS_IS_LV2_TURTLE_SCANNER()
#define	AGS_IS_LV2_TURTLE_SCANNER_CLASS()
#define	AGS_LV2_TURTLE_SCANNER()
#define	AGS_LV2_TURTLE_SCANNER_CLASS()
#define	AGS_LV2_TURTLE_SCANNER_GET_CLASS()
GType	ags_lv2_turtle_scanner_get_type ()
GType	ags_lv2_turtle_scanner_status_flags_get_type ()

Types and Values

enum	AgsLv2TurtleScannerStatusFlags
struct	AgsLv2CacheTurtle
#define	AGS_TYPE_LV2_TURTLE_SCANNER
struct	AgsLv2TurtleScanner
struct	AgsLv2TurtleScannerClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsLv2TurtleScannerStatusFlags
GObject
&#x2570;&#x2500;&#x2500; AgsLv2TurtleScanner
```

Includes

```
#include <ags/plugin/ags_lv2_turtle_scanner.h>
```

Description

The AgsLv2TurtleScanner parses RDF Turtle files.

Functions

AGS_LV2_CACHE_TURTLE()

```
#define AGS_LV2_CACHE_TURTLE(ptr) ((AgsLv2CacheTurtle *) ptr)
```

AGS_LV2_TURTLE_SCANNER_GET_OBJ_MUTEX()

```
#define AGS_LV2_TURTLE_SCANNER_GET_OBJ_MUTEX(obj) (&(((AgsLv2TurtleScanner *) obj)-> ↩
    obj_mutex))
```

ags_lv2_cache_turtle_alloc ()

```
AgsLv2CacheTurtle~*
ags_lv2_cache_turtle_alloc (AgsLv2CacheTurtle *parent,
                            gchar *turtle_filename);
```

Allocate an AgsLv2CacheTurtle

Parameters

parent	the parent AgsLv2CacheTurtle	
turtle_filename	the .ttl filename	

Returns

a new AgsLv2CacheTurtle

Since: 3.2.7

ags_lv2_cache_turtle_free ()

```
void
ags_lv2_cache_turtle_free (AgsLv2CacheTurtle *lv2_cache_turtle);
```

Free an AgsLv2CacheTurtle

Parameters

lv2_cache_turtle	the AgsLv2CacheTurtle	

Since: 3.2.7

ags_lv2_cache_turtle_find ()

```
GList~*
ags_lv2_cache_turtle_find (GList *lv2_cache_turtle,
                           gchar *turtle_filename);
```

Find *lv2_cache_turtle* matching *turtle_filename* .

Parameters

lv2_cache_turtle	a GList containing AgsLv2CacheTurtle	
turtle_filename	the .ttl filename	

Returns

the next matching GList

Since: 3.2.11

ags_lv2_turtle_scanner_quick_scan_see_also ()

```
void
ags_lv2_turtle_scanner_quick_scan_see_also
                                (AgsLv2TurtleScanner *lv2_turtle_scanner,
                                 AgsLv2CacheTurtle *parent,
                                 gchar *turtle_filename);
```

ags_lv2_turtle_scanner_quick_scan ()

```
void
ags_lv2_turtle_scanner_quick_scan (AgsLv2TurtleScanner *lv2_turtle_scanner,
                                   gchar *manifest_filename);
```

Quick scan to detect available plugins.

Parameters

lv2_turtle_scanner	the AgsLv2TurtleScanner	
manifest_filename	the manifest filename	

Since: 3.2.7

ags_lv2_turtle_scanner_new ()

```
AgsLv2TurtleScanner~*
ags_lv2_turtle_scanner_new ();
```

Creates an AgsLv2TurtleScanner

Returns

a new AgsLv2TurtleScanner

Since: 3.2.7

AGS_IS_LV2_TURTLE_SCANNER()

```
#define AGS_IS_LV2_TURTLE_SCANNER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_TURTLE_SCANNER))
```

AGS_IS_LV2_TURTLE_SCANNER_CLASS()

```
#define AGS_IS_LV2_TURTLE_SCANNER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_TURTLE_SCANNER))
```

AGS_LV2_TURTLE_SCANNER()

```
#define AGS_LV2_TURTLE_SCANNER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_TURTLE_SCANNER, AgsLv2TurtleScanner))
```

AGS_LV2_TURTLE_SCANNER_CLASS()

```
#define AGS_LV2_TURTLE_SCANNER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_TURTLE_SCANNER, AgsLv2TurtleScannerClass))
```

AGS_LV2_TURTLE_SCANNER_GET_CLASS()

```
#define AGS_LV2_TURTLE_SCANNER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_TURTLE_SCANNER, AgsLv2TurtleScannerClass))
```

ags_lv2_turtle_scanner_get_type ()

```
GType
ags_lv2_turtle_scanner_get_type (void);
```

ags_lv2_turtle_scanner_status_flags_get_type ()

```
GType
ags_lv2_turtle_scanner_status_flags_get_type
                                 ();
```

Types and Values

enum AgsLv2TurtleScannerStatusFlags

Enum values to indicate internal state of AgsLv2TurtleScanner by enable/disable as status flags.

Members

AGS_LV2_TURTLE_SCANNER_A	scanner a verb
AGS_LV2_TURTLE_SCANNER_SEEALSO	scanner see also
AGS_LV2_TURTLE_SCANNER_BINARY	scanner bi- nary
AGS_LV2_TURTLE_SCANNER_NAME	scanner name

struct AgsLv2CacheTurtle

```
struct AgsLv2CacheTurtle {
  AgsLv2CacheTurtle *parent;

  gchar *turtle_filename;

  GHashTable *prefix_id;

  GHashTable *plugin_filename;
  GHashTable *plugin_effect;

  GHashTable *is_plugin;
  GHashTable *is_instrument;

  guint plugin_count;

  gchar **see_also;
};
```

AGS_TYPE_LV2_TURTLE_SCANNER

```
#define AGS_TYPE_LV2_TURTLE_SCANNER                (ags_lv2_turtle_scanner_get_type())
```

struct AgsLv2TurtleScanner

```
struct AgsLv2TurtleScanner;
```

struct AgsLv2TurtleScannerClass

```
struct AgsLv2TurtleScannerClass {
  GObjectClass gobject;
};
```

4.8 AgsLv2Plugin

AgsLv2Plugin — The lv2 plugin class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_PLUGIN_DESCRIPTOR()
#define	AGS_LV2_EVENT_BUFFER()
#define	AGS_LV2_EVENT()
#define	AGS_LV2_EVENT_DATA()
#define	AGS_LV2_ATOM_SEQUENCE()
#define	AGS_LV2_ATOM_EVENT()
gboolean	ags_lv2_plugin_test_flags ()
void	ags_lv2_plugin_set_flags ()
void	ags_lv2_plugin_unset_flags ()
LV2_Event_Buffer *	ags_lv2_plugin_event_buffer_alloc ()
void	ags_lv2_plugin_event_buffer_free ()
void	ags_lv2_plugin_event_buffer_realloc_data ()
LV2_Event_Buffer *	ags_lv2_plugin_event_buffer_concat ()
gboolean	ags_lv2_plugin_event_buffer_append_midi ()
gboolean	ags_lv2_plugin_event_buffer_remove_midi ()
void	ags_lv2_plugin_clear_event_buffer ()
gpointer	ags_lv2_plugin_alloc_atom_sequence ()
void	ags_lv2_plugin_atom_sequence_free ()
gboolean	ags_lv2_plugin_atom_sequence_append_midi ()
gboolean	ags_lv2_plugin_atom_sequence_remove_midi ()
void	ags_lv2_plugin_clear_atom_sequence ()
GList *	ags_lv2_plugin_find_uri ()
GList *	ags_lv2_plugin_find_pname ()
void	ags_lv2_plugin_change_program ()
AgsLv2Plugin *	ags_lv2_plugin_new ()
#define	AGS_IS_LV2_PLUGIN()
#define	AGS_IS_LV2_PLUGIN_CLASS()
#define	AGS_LV2_PLUGIN()
#define	AGS_LV2_PLUGIN_CLASS()
#define	AGS_LV2_PLUGIN_GET_CLASS()
GType	ags_lv2_plugin_flags_get_type ()
GType	ags_lv2_plugin_get_type ()

Properties

char *	doap-name	Read / Write
char *	foaf-homepage	Read / Write
char *	foaf-mbox	Read / Write
char *	foaf-name	Read / Write
AgsTurtle *	manifest	Read / Write
char *	pname	Read / Write
gpointer	preset	Read / Write
AgsTurtle *	turtle	Read / Write
char *	ui-uri	Read / Write
char *	uri	Read / Write

Signals

void	change-program	Run Last

Types and Values

enum	AgsLv2PluginFlags
#define	AGS_TYPE_LV2_PLUGIN
struct	AgsLv2Plugin
struct	AgsLv2PluginClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsLv2PluginFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsBasePlugin
        &#x2570;&#x2500;&#x2500; AgsLv2Plugin
```

Includes

```
#include <ags/plugin/ags_lv2_plugin.h>
```

Description

The AgsLv2Plugin loads/unloads a Lv2 plugin.

Functions

AGS_LV2_PLUGIN_DESCRIPTOR()

```
#define AGS_LV2_PLUGIN_DESCRIPTOR(ptr) ((LV2_Descriptor *)(ptr))
```

AGS_LV2_EVENT_BUFFER()

```
#define AGS_LV2_EVENT_BUFFER(ptr) ((LV2_Event_Buffer *)(ptr))
```

AGS_LV2_EVENT()

```
#define AGS_LV2_EVENT(ptr) ((LV2_Event *)(ptr))
```

AGS_LV2_EVENT_DATA()

```
#define AGS_LV2_EVENT_DATA(ptr) ((void *)(ptr + sizeof(LV2_Event)))
```

AGS_LV2_ATOM_SEQUENCE()

```
#define AGS_LV2_ATOM_SEQUENCE(ptr) ((LV2_Atom_Sequence *)(ptr))
```

AGS_LV2_ATOM_EVENT()

```
#define AGS_LV2_ATOM_EVENT(ptr) ((LV2_Atom_Event *)(ptr))
```

ags_lv2_plugin_test_flags ()

```
gboolean
ags_lv2_plugin_test_flags (AgsLv2Plugin *lv2_plugin,
                           guint flags);
```

Test *flags* to be set on *recall* .

Parameters

lv2_plugin	the AgsLv2Plugin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_lv2_plugin_set_flags ()

```
void
ags_lv2_plugin_set_flags (AgsLv2Plugin *lv2_plugin,
                          guint flags);
```

Set flags.

Parameters

lv2_plugin	the AgsLv2Plugin	
flags	the flags	

Since: 3.0.0

ags_lv2_plugin_unset_flags ()

```
void
ags_lv2_plugin_unset_flags (AgsLv2Plugin *lv2_plugin,
                            guint flags);
```

Unset flags.

Parameters

lv2_plugin	the AgsLv2Plugin	
flags	the flags	

Since: 3.0.0

ags_lv2_plugin_event_buffer_alloc ()

```
LV2_Event_Buffer~*
ags_lv2_plugin_event_buffer_alloc (guint buffer_size);
```

Allocate LV2_Event_Buffer struct.

Parameters

buffer_size	the data's buffer size	

Returns

a new allocated LV2_Event_Buffer.

[type gpointer][transfer none]

Since: 3.0.0

ags_lv2_plugin_event_buffer_free ()

```
void
ags_lv2_plugin_event_buffer_free (gpointer event_buffer);
```

Free LV2_Event_Buffer struct.

Parameters

event_buffer	the LV2_Event_Buffer struct.	*[type gpointer][transfer none]*

Since: 3.3.0

ags_lv2_plugin_event_buffer_realloc_data ()

```
void
ags_lv2_plugin_event_buffer_realloc_data
                                (LV2_Event_Buffer *event_buffer,
                                 guint buffer_size);
```

Reallocate LV2_Event_Buffer struct's data field.

Parameters

event_buffer	the LV2_Event_Buffer struct.	*[type gpointer][transfer none]*
buffer_size	the data's buffer size	

Since: 3.0.0

ags_lv2_plugin_event_buffer_concat ()

```
LV2_Event_Buffer~*
ags_lv2_plugin_event_buffer_concat (LV2_Event_Buffer *event_buffer,
                                    ...);
```

Concats the event buffers.

Parameters

event_buffer	the first buffer.	*[type gpointer][transfer none]*
...	NULL terminated variadict arguments	

Returns

The newly allocated event buffer.

[type gpointer][transfer none]

Since: 3.0.0

ags_lv2_plugin_event_buffer_append_midi ()

```
gboolean
ags_lv2_plugin_event_buffer_append_midi
                                (gpointer event_buffer,
                                 guint buffer_size,
                                 snd_seq_event_t *events,
                                 guint event_count);
```

Append MIDI data to event buffer.

Parameters

event_buffer	the event buffer	
buffer_size	the event buffer size	
events	the events to write.	*[type gpointer][transfer none]*
event_count	the number of events to write	

Returns

TRUE on success otherwise FALSE

Since: 3.0.0

ags_lv2_plugin_event_buffer_remove_midi ()

```
gboolean
ags_lv2_plugin_event_buffer_remove_midi
                                (gpointer event_buffer,
                                 guint buffer_size,
                                 guint note);
```

Remove MIDI data from event buffer.

Parameters

event_buffer	the event buffer	
buffer_size	the event buffer size	
note	the key to remove	

Returns

TRUE on success otherwise FALSE

Since: 3.0.0

ags_lv2_plugin_clear_event_buffer ()

```
void
ags_lv2_plugin_clear_event_buffer (gpointer event_buffer,
                                   guint buffer_size);
```

Clear the event buffer.

Parameters

event_buffer	the event buffer	
buffer_size	size of *event_buffer*	

Since: 3.0.0

ags_lv2_plugin_alloc_atom_sequence ()

```
gpointer
ags_lv2_plugin_alloc_atom_sequence (guint sequence_size);
```

Allocates a LV2_Atom_Sequence

Parameters

sequence_size	the requested size	

Returns

the new atom sequence

Since: 3.0.0

ags_lv2_plugin_atom_sequence_free ()

```
void
ags_lv2_plugin_atom_sequence_free (gpointer atom_sequence);
```

Free atom sequence.

Parameters

atom_sequence	the atom sequence	

Since: 3.3.0

ags_lv2_plugin_atom_sequence_append_midi ()

```
gboolean
ags_lv2_plugin_atom_sequence_append_midi
                                (gpointer atom_sequence,
                                 guint sequence_size,
                                 snd_seq_event_t *events,
                                 guint event_count);
```

Append MIDI data to atom sequence.

Parameters

atom_sequence	the atom sequence	
sequence_size	the atom sequence size	
events	the events to write.	*[type gpointer][transfer none]*
event_count	the number of events to write	

Returns

TRUE on success otherwise FALSE

Since: 3.0.0

ags_lv2_plugin_atom_sequence_remove_midi ()

```
gboolean
ags_lv2_plugin_atom_sequence_remove_midi
                                (gpointer atom_sequence,
                                 guint sequence_size,
                                 guint note);
```

Remove MIDI data from atom sequence.

Parameters

atom_sequence	the atom sequence	
sequence_size	the atom sequence size	
note	the key to remove	

Returns

TRUE on success otherwise FALSE

Since: 3.0.0

ags_lv2_plugin_clear_atom_sequence ()

```
void
ags_lv2_plugin_clear_atom_sequence (gpointer atom_sequence,
                                    guint sequence_size);
```

Clear the atom sequence.

Parameters

atom_sequence	the atom sequence	
sequence_size	size of *atom_sequence*	

Since: 3.0.0

ags_lv2_plugin_find_uri ()

```
GList~*
ags_lv2_plugin_find_uri (GList *lv2_plugin,
                         gchar *uri);
```

Find uri in *lv2_plugin* GList

Parameters

lv2_plugin	a GList containig AgsLv2Plugin.	*[element-type AgsAudio.Lv2Plugin][transfer none]*
uri	the uri to find	

Returns

the matching GList containing AgsLv2Plugin.

[element-type AgsAudio.Lv2Plugin][transfer none]

Since: 3.0.0

ags_lv2_plugin_find_pname ()

```
GList~*
ags_lv2_plugin_find_pname (GList *lv2_plugin,
                           gchar *pname);
```

Find pname in *lv2_plugin* GList

Parameters

lv2_plugin	a GList containig AgsLv2Plugin.	*[element-type AgsAudio.Lv2Plugin][transfer none]*
pname	the pname to find	

Returns

the matching GList containing AgsLv2Plugin.

[element-type AgsAudio.Lv2Plugin][transfer none]

Since: 3.0.0

ags_lv2_plugin_change_program ()

```
void
ags_lv2_plugin_change_program (AgsLv2Plugin *lv2_plugin,
                               gpointer lv2_handle,
                               guint bank_index,
                               guint program_index);
```

Change program of `lv2_handle`.

Parameters

lv2_plugin	the AgsLv2Plugin	
lv2_handle	the lv2 handle	
bank_index	the bank index	
program_index	the program index	

Since: 3.0.0

ags_lv2_plugin_new ()

```
AgsLv2Plugin~*
ags_lv2_plugin_new (AgsTurtle *turtle,
                    gchar *filename,
                    gchar *effect,
                    gchar *uri,
                    guint effect_index);
```

Creates an AgsLv2Plugin

Parameters

turtle	the AgsTurtle	
filename	the plugin .so	
effect	the effect's string representation	
uri	the effect's uri	
effect_index	the effect's index	

Returns

a new AgsLv2Plugin

Since: 3.0.0

AGS_IS_LV2_PLUGIN()

```
#define AGS_IS_LV2_PLUGIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_PLUGIN))
```

AGS_IS_LV2_PLUGIN_CLASS()

```
#define AGS_IS_LV2_PLUGIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_PLUGIN))
```

AGS_LV2_PLUGIN()

```
#define AGS_LV2_PLUGIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_PLUGIN, AgsLv2Plugin))
```

AGS_LV2_PLUGIN_CLASS()

```
#define AGS_LV2_PLUGIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_PLUGIN, AgsLv2PluginClass))
```

AGS_LV2_PLUGIN_GET_CLASS()

```
#define AGS_LV2_PLUGIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_PLUGIN, AgsLv2PluginClass))
```

ags_lv2_plugin_flags_get_type ()

```
GType
ags_lv2_plugin_flags_get_type ();
```

ags_lv2_plugin_get_type ()

```
GType
ags_lv2_plugin_get_type (void);
```

Types and Values

enum AgsLv2PluginFlags

Enum values to control the behavior or indicate internal state of AgsLv2Plugin by enable/disable as flags.

Members

AGS_LV2_PLUGIN_IS_SYNTHESIZER	plugin is synthesizer
AGS_LV2_PLUGIN_NEEDS_WORKER	plugin needs worker
AGS_LV2_PLUGIN_HAS_PROGRAM_INTERFACE	plugin has program interface

AGS_TYPE_LV2_PLUGIN

```
#define AGS_TYPE_LV2_PLUGIN                (ags_lv2_plugin_get_type())
```

struct AgsLv2Plugin

```
struct AgsLv2Plugin;
```

struct AgsLv2PluginClass

```
struct AgsLv2PluginClass {
  AgsBasePluginClass base_plugin;

  void (*change_program)(AgsLv2Plugin *lv2_plugin,
       gpointer lv2_handle,
       guint bank_index,
       guint program_index);
};
```

Property Details

The "doap-name" property

```
  "doap-name"                char~*
```

The assigned doap name.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "foaf-homepage" property

```
"foaf-homepage"            char~*
```

The assigned foaf homepage.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "foaf-mbox" property

```
"foaf-mbox"                char~*
```

The assigned foaf mbox.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "foaf-name" property

```
"foaf-name"                char~*
```

The assigned foaf name.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "manifest" property

```
"manifest"                 AgsTurtle~*
```

The assigned manifest.

Owner: AgsLv2Plugin

Flags: Read / Write

Since: 3.0.0

The "pname" property

```
"pname"                    char~*
```

The assigned pname.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "preset" property

```
"preset"                    gpointer
```

The assigned preset.

[transfer full]

Owner: AgsLv2Plugin

Flags: Read / Write

Since: 3.0.0

The "turtle" property

```
"turtle"                    AgsTurtle~*
```

The assigned turtle.

Owner: AgsLv2Plugin

Flags: Read / Write

Since: 3.0.0

The "ui-uri" property

```
"ui-uri"                    char~*
```

The assigned ui-uri.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "uri" property

```
"uri"                       char~*
```

The assigned uri.

Owner: AgsLv2Plugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Signal Details

The "change-program" signal

```
void
user_function (AgsLv2Plugin *lv2_plugin,
               gpointer      lv2_handle,
               guint         bank,
               guint         program,
               gpointer      user_data)
```

The ::change-program signal creates a new instance of plugin.

Parameters

lv2_plugin	the plugin to change-program	
lv2_handle	the Lv2 handle	
bank	the bank number	
program	the program number	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

4.9 AgsLv2PresetManager

AgsLv2PresetManager — Singleton pattern to organize LV2 presets

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_PRESET_MANAGER_GET_OBJ_MUTEX()
AgsLv2PresetManager *	ags_lv2_preset_manager_get_instance ()
AgsLv2PresetManager *	ags_lv2_preset_manager_new ()
#define	AGS_IS_LV2_PRESET_MANAGER()
#define	AGS_IS_LV2_PRESET_MANAGER_CLASS()
#define	AGS_LV2_PRESET_MANAGER()
#define	AGS_LV2_PRESET_MANAGER_CLASS()
#define	AGS_LV2_PRESET_MANAGER_GET_CLASS()
GType	ags_lv2_preset_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2_PRESET_MANAGER
struct	AgsLv2PresetManager
struct	AgsLv2PresetManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsLv2PresetManager
```

Includes

```
#include <ags/plugin/ags_lv2_preset_manager.h>
```

Description

The AgsLv2PresetManager loads/unloads LV2 presets.

Functions

AGS_LV2_PRESET_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2_PRESET_MANAGER_GET_OBJ_MUTEX(obj) (((AgsLv2PresetManager *) obj)->obj_mutex ↩
    )
```

ags_lv2_preset_manager_get_instance ()

```
AgsLv2PresetManager~*
ags_lv2_preset_manager_get_instance ();
```

Get instance.

Returns

the AgsLv2PresetManager.

[transfer none]

Since: 3.0.0

ags_lv2_preset_manager_new ()

```
AgsLv2PresetManager~*
ags_lv2_preset_manager_new ();
```

Create a new instance of AgsLv2PresetManager

Returns

the new AgsLv2PresetManager

Since: 3.0.0

AGS_IS_LV2_PRESET_MANAGER()

```
#define AGS_IS_LV2_PRESET_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_PRESET_MANAGER))
```

AGS_IS_LV2_PRESET_MANAGER_CLASS()

```
#define AGS_IS_LV2_PRESET_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_PRESET_MANAGER))
```

AGS_LV2_PRESET_MANAGER()

```
#define AGS_LV2_PRESET_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_PRESET_MANAGER, AgsLv2PresetManager))
```

AGS_LV2_PRESET_MANAGER_CLASS()

```
#define AGS_LV2_PRESET_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_PRESET_MANAGER, AgsLv2PresetManagerClass))
```

AGS_LV2_PRESET_MANAGER_GET_CLASS()

```
#define AGS_LV2_PRESET_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_PRESET_MANAGER, AgsLv2PresetManagerClass))
```

ags_lv2_preset_manager_get_type ()

```
GType
ags_lv2_preset_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2_PRESET_MANAGER

```
#define AGS_TYPE_LV2_PRESET_MANAGER                 (ags_lv2_preset_manager_get_type())
```

struct AgsLv2PresetManager

```
struct AgsLv2PresetManager;
```

struct AgsLv2PresetManagerClass

```
struct AgsLv2PresetManagerClass {
  GObjectClass gobject;
};
```

4.10 AgsLv2Preset

AgsLv2Preset — LV2 presets

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_PRESET_GET_OBJ_MUTEX()
#define	AGS_LV2_PORT_PRESET()
AgsLv2PortPreset *	ags_lv2_port_preset_alloc ()
void	ags_lv2_port_preset_free ()
void	ags_lv2_preset_parse_turtle ()
GList *	ags_lv2_preset_find_preset_uri ()
GList *	ags_lv2_preset_find_preset_label ()
AgsLv2Preset *	ags_lv2_preset_new ()
#define	AGS_IS_LV2_PRESET()
#define	AGS_IS_LV2_PRESET_CLASS()
#define	AGS_LV2_PRESET()
#define	AGS_LV2_PRESET_CLASS()
#define	AGS_LV2_PRESET_GET_CLASS()
GType	ags_lv2_preset_get_type ()

Properties

char *	applies-to	Read / Write
char *	bank	Read / Write
AgsLv2Plugin *	lv2-plugin	Read / Write
AgsTurtle *	manifest	Read / Write
gpointer	port-preset	Read
char *	preset-label	Read / Write
AgsTurtle *	turtle	Read / Write
char *	uri	Read / Write

Types and Values

struct	AgsLv2PortPreset
#define	AGS_TYPE_LV2_PRESET
struct	AgsLv2Preset
struct	AgsLv2PresetClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLv2Preset
```

Includes

```
#include <ags/plugin/ags_lv2_preset.h>
```

Description

The AgsLv2Preset handles LV2 presets mapped to a bank.

Functions

AGS_LV2_PRESET_GET_OBJ_MUTEX()

```
#define AGS_LV2_PRESET_GET_OBJ_MUTEX(obj) (&(((AgsLv2Preset *) obj)->obj_mutex))
```

AGS_LV2_PORT_PRESET()

```
#define AGS_LV2_PORT_PRESET(ptr) ((AgsLv2PortPreset*)(ptr))
```

ags_lv2_port_preset_alloc ()

```
AgsLv2PortPreset~*
ags_lv2_port_preset_alloc (gchar *port_symbol,
                           GType port_type);
```

Allocated AgsLv2PortPreset.

Parameters

port_symbol	the port symbol	
port_type	the port type	

Returns

the new AgsLv2PortPreset.

[type gpointer][transfer none]

Since: 3.0.0

ags_lv2_port_preset_free ()

```
void
ags_lv2_port_preset_free (AgsLv2PortPreset *lv2_port_preset);
```

Free *lv2_port_preset* .

Parameters

lv2_port_preset	the AgsLv2PortPreset.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_lv2_preset_parse_turtle ()

```
void
ags_lv2_preset_parse_turtle (AgsLv2Preset *lv2_preset);
```

Parse *lv2_preset* .

Parameters

lv2_preset	the AgsLv2Preset	

Since: 3.0.0

ags_lv2_preset_find_preset_uri ()

```
GList~*
ags_lv2_preset_find_preset_uri (GList *lv2_preset,
                                gchar *preset_uri);
```

Find *preset_uri* within *lv2_preset* .

Parameters

lv2_preset	the GList containing AgsLv2Preset.	*[element-type AgsAudio.Lv2Preset][transfer none]*
preset_uri	the preset URI	

Returns

the matching GList containing AgsLv2Preset.

[element-type AgsAudio.Lv2Preset][transfer none]

Since: 3.0.0

ags_lv2_preset_find_preset_label ()

```
GList~*
ags_lv2_preset_find_preset_label (GList *lv2_preset,
                                  gchar *preset_label);
```

Find *preset_label* within *lv2_preset* .

Parameters

lv2_preset	the GList containing AgsLv2Preset.	*[element-type AgsAudio.Lv2Preset][transfer none]*
preset_label	the preset label	

Returns

the matching GList containing AgsLv2Preset.

[element-type AgsAudio.Lv2Preset][transfer none]

Since: 3.0.0

ags_lv2_preset_new ()

```
AgsLv2Preset~*
ags_lv2_preset_new (GObject *lv2_plugin,
                    AgsTurtle *turtle,
                    gchar *uri);
```

Create a new instance of AgsLv2Preset.

Parameters

lv2_plugin	an AgsLv2Plugin	
turtle	the AgsTurtle	
uri	the URI as string	

Returns

the new AgsLv2Preset

Since: 3.0.0

AGS_IS_LV2_PRESET()

```
#define AGS_IS_LV2_PRESET(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_PRESET))
```

AGS_IS_LV2_PRESET_CLASS()

```
#define AGS_IS_LV2_PRESET_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_PRESET))
```

AGS_LV2_PRESET()

```
#define AGS_LV2_PRESET(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_PRESET, AgsLv2Preset))
```

AGS_LV2_PRESET_CLASS()

```
#define AGS_LV2_PRESET_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_PRESET, AgsLv2PresetClass))
```

AGS_LV2_PRESET_GET_CLASS()

```
#define AGS_LV2_PRESET_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_PRESET, AgsLv2PresetClass))
```

ags_lv2_preset_get_type ()

```
GType
ags_lv2_preset_get_type (void);
```

Types and Values

struct AgsLv2PortPreset

```
struct AgsLv2PortPreset {
  gchar *port_symbol;

  GValue *port_value;
};
```

AGS_TYPE_LV2_PRESET

```
#define AGS_TYPE_LV2_PRESET                (ags_lv2_preset_get_type())
```

struct AgsLv2Preset

```
struct AgsLv2Preset;
```

struct AgsLv2PresetClass

```
struct AgsLv2PresetClass {
  GObjectClass gobject;
};
```

Property Details

The "applies-to" property

```
  "applies-to"               char~*
```

The assigned applies to.

Owner: AgsLv2Preset

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "bank" property

```
  "bank"                     char~*
```

The assigned bank.

Owner: AgsLv2Preset

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "lv2-plugin" property

```
"lv2-plugin"                AgsLv2Plugin~*
```

The assigned lv2 plugin.

Owner: AgsLv2Preset

Flags: Read / Write

Since: 3.0.0

The "manifest" property

```
"manifest"                  AgsTurtle~*
```

The assigned manifest.

Owner: AgsLv2Preset

Flags: Read / Write

Since: 3.0.0

The "port-preset" property

```
"port-preset"               gpointer
```

The port preset.

Owner: AgsLv2Preset

Flags: Read

Since: 3.0.0

The "preset-label" property

```
"preset-label"              char~*
```

The preset label.

Owner: AgsLv2Preset

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "turtle" property

```
"turtle"                    AgsTurtle~*
```

The assigned turtle.

Owner: AgsLv2Preset

Flags: Read / Write

Since: 3.0.0

The "uri" property

```
"uri"                          char~*
```

The assigned uri.

Owner: AgsLv2Preset

Flags: Read / Write

Default value: NULL

Since: 3.0.0

4.11 AgsLv2UriMapManager

AgsLv2UriMapManager — lv2 uri map manager

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_URI_MAP_MANAGER_GET_OBJ_MUTEX()
gboolean	ags_lv2_uri_map_manager_insert ()
gboolean	ags_lv2_uri_map_manager_remove ()
GValue *	ags_lv2_uri_map_manager_lookup ()
void	ags_lv2_uri_map_manager_load_default ()
uint32_t	ags_lv2_uri_map_manager_uri_to_id ()
AgsLv2UriMapManager *	ags_lv2_uri_map_manager_get_instance ()
AgsLv2UriMapManager *	ags_lv2_uri_map_manager_new ()
#define	AGS_IS_LV2_URI_MAP_MANAGER()
#define	AGS_IS_LV2_URI_MAP_MANAGER_CLASS()
#define	AGS_LV2_URI_MAP_MANAGER()
#define	AGS_LV2_URI_MAP_MANAGER_CLASS()
#define	AGS_LV2_URI_MAP_MANAGER_GET_CLASS()
GType	ags_lv2_uri_map_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2_URI_MAP_MANAGER
struct	AgsLv2UriMapManager
struct	AgsLv2UriMapManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLv2UriMapManager
```

Includes

```
#include <ags/plugin/ags_lv2_uri_map_manager.h>
```

Description

The AgsLv2UriMapManager keeps your uri in a hash table where you can lookup your ids.

Functions

AGS_LV2_URI_MAP_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2_URI_MAP_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLv2UriMapManager *) obj)-> ↩
    obj_mutex))
```

ags_lv2_uri_map_manager_insert ()

```
gboolean
ags_lv2_uri_map_manager_insert (AgsLv2UriMapManager *lv2_uri_map_manager,
                                gchar *uri,
                                GValue *id);
```

Inserts a id into hash map associated with *uri* .

Parameters

lv2_uri_map_manager	the AgsLv2UriMapManager	
uri	the object as key	
id	the id to insert	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_lv2_uri_map_manager_remove ()

```
gboolean
ags_lv2_uri_map_manager_remove (AgsLv2UriMapManager *lv2_uri_map_manager,
                                gchar *uri);
```

Removes a lock associated with *uri*.3

Parameters

lv2_uri_map_manager	the AgsLv2UriMapManager	
uri	the object to remove	

Returns

TRUE as successfully removed, otherwise FALSE

Since: 3.0.0

ags_lv2_uri_map_manager_lookup ()

```
GValue~*
ags_lv2_uri_map_manager_lookup (AgsLv2UriMapManager *lv2_uri_map_manager,
                                gchar *uri);
```

Lookup a id associated with *uri* in *lv2_uri_map_manager*

Parameters

lv2_uri_map_manager	the AgsLv2UriMapManager	
uri	the object to lookup	

Returns

the id on success, else G_MAXUINT32

Since: 3.0.0

ags_lv2_uri_map_manager_load_default ()

```
void
ags_lv2_uri_map_manager_load_default (AgsLv2UriMapManager *lv2_uri_map_manager);
```

ags_lv2_uri_map_manager_uri_to_id ()

```
uint32_t
ags_lv2_uri_map_manager_uri_to_id (LV2_URI_Map_Callback_Data callback_data,
                                   char *map,
                                   char *uri);
```

ags_lv2_uri_map_manager_get_instance ()

```
AgsLv2UriMapManager~*
ags_lv2_uri_map_manager_get_instance ();
```

Singleton function to optain the id manager instance.

Returns

an instance of AgsLv2UriMapManager.

[transfer none]

Since: 3.0.0

ags_lv2_uri_map_manager_new ()

```
AgsLv2UriMapManager~*
ags_lv2_uri_map_manager_new ();
```

Instantiate a id manager.

Returns

a new AgsLv2UriMapManager

Since: 3.0.0

AGS_IS_LV2_URI_MAP_MANAGER()

```
#define AGS_IS_LV2_URI_MAP_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_URI_MAP_MANAGER))
```

AGS_IS_LV2_URI_MAP_MANAGER_CLASS()

```
#define AGS_IS_LV2_URI_MAP_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_URI_MAP_MANAGER))
```

AGS_LV2_URI_MAP_MANAGER()

```
#define AGS_LV2_URI_MAP_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_URI_MAP_MANAGER, AgsLv2UriMapManager))
```

AGS_LV2_URI_MAP_MANAGER_CLASS()

```
#define AGS_LV2_URI_MAP_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_URI_MAP_MANAGER, AgsLv2UriMapManagerClass))
```

AGS_LV2_URI_MAP_MANAGER_GET_CLASS()

```
#define AGS_LV2_URI_MAP_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_URI_MAP_MANAGER, AgsLv2UriMapManagerClass))
```

ags_lv2_uri_map_manager_get_type ()

```
GType
ags_lv2_uri_map_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2_URI_MAP_MANAGER

```
#define AGS_TYPE_LV2_URI_MAP_MANAGER                (ags_lv2_uri_map_manager_get_type())
```

struct AgsLv2UriMapManager

```
struct AgsLv2UriMapManager;
```

struct AgsLv2UriMapManagerClass

```
struct AgsLv2UriMapManagerClass {
  GObjectClass gobject;
};
```

4.12 AgsLv2UridManager

AgsLv2UridManager — LV2 urid manager

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_URID_MANAGER_GET_OBJ_MUTEX()
gboolean	ags_lv2_urid_manager_insert ()
gboolean	ags_lv2_urid_manager_remove ()
GValue *	ags_lv2_urid_manager_lookup ()
void	ags_lv2_urid_manager_load_default ()
uint32_t	ags_lv2_urid_manager_map ()
const char *	ags_lv2_urid_manager_unmap ()
AgsLv2UridManager *	ags_lv2_urid_manager_get_instance ()
AgsLv2UridManager *	ags_lv2_urid_manager_new ()
#define	AGS_IS_LV2_URID_MANAGER()
#define	AGS_IS_LV2_URID_MANAGER_CLASS()
#define	AGS_LV2_URID_MANAGER()
#define	AGS_LV2_URID_MANAGER_CLASS()
#define	AGS_LV2_URID_MANAGER_GET_CLASS()
GType	ags_lv2_urid_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2_URID_MANAGER
struct	AgsLv2UridManager
struct	AgsLv2UridManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsLv2UridManager
```

Includes

```
#include <ags/plugin/ags_lv2_urid_manager.h>
```

Description

The AgsLv2UridManager keeps your urid in a hash table where you can lookup your ids.

Functions

AGS_LV2_URID_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2_URID_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLv2UridManager *) obj)->obj_mutex))
```

ags_lv2_urid_manager_insert ()

```
gboolean
ags_lv2_urid_manager_insert (AgsLv2UridManager *lv2_urid_manager,
                             gchar *uri,
                             GValue *id);
```

Inserts a id into hash associated with *urid* .

Parameters

lv2_urid_manager	the AgsLv2UridManager	
uri	the object as key	
id	the id to insert	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_lv2_urid_manager_remove ()

```
gboolean
ags_lv2_urid_manager_remove (AgsLv2UridManager *lv2_urid_manager,
                             gchar *uri);
```

Removes an entry associated with *urid* .

Parameters

lv2_urid_manager	the AgsLv2UridManager	
uri	the object to remove	

Returns

TRUE as successfully removed, otherwise FALSE

Since: 3.0.0

ags_lv2_urid_manager_lookup ()

```
GValue~*
ags_lv2_urid_manager_lookup (AgsLv2UridManager *lv2_urid_manager,
                             gchar *uri);
```

Lookup a id associated with *urid* in *lv2_urid_manager*

Parameters

lv2_urid_manager	the AgsLv2UridManager	
uri	the object to lookup	

Returns

the id on success, else the newly created id

Since: 3.0.0

ags_lv2_urid_manager_load_default ()

```
void
ags_lv2_urid_manager_load_default (AgsLv2UridManager *lv2_urid_manager);
```

ags_lv2_urid_manager_map ()

```
uint32_t
ags_lv2_urid_manager_map (LV2_URID_Map_Handle handle,
                          char *uri);
```

ags_lv2_urid_manager_unmap ()

```
const char~*
ags_lv2_urid_manager_unmap (LV2_URID_Map_Handle handle,
                            uint32_t urid);
```

ags_lv2_urid_manager_get_instance ()

```
AgsLv2UridManager~*
ags_lv2_urid_manager_get_instance ();
```

Singleton function to optain the id manager instance.

Returns

an instance of AgsLv2UridManager.

[transfer none]

Since: 3.0.0

ags_lv2_urid_manager_new ()

```
AgsLv2UridManager~*
ags_lv2_urid_manager_new ();
```

Instantiate a id manager.

Returns

a new AgsLv2UridManager

Since: 3.0.0

AGS_IS_LV2_URID_MANAGER()

```
#define AGS_IS_LV2_URID_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_URID_MANAGER))
```

AGS_IS_LV2_URID_MANAGER_CLASS()

```
#define AGS_IS_LV2_URID_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_URID_MANAGER))
```

AGS_LV2_URID_MANAGER()

```
#define AGS_LV2_URID_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_URID_MANAGER, AgsLv2UridManager))
```

AGS_LV2_URID_MANAGER_CLASS()

```
#define AGS_LV2_URID_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_URID_MANAGER, AgsLv2UridManagerClass))
```

AGS_LV2_URID_MANAGER_GET_CLASS()

```
#define AGS_LV2_URID_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_URID_MANAGER, AgsLv2UridManagerClass))
```

ags_lv2_urid_manager_get_type ()

```
GType
ags_lv2_urid_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2_URID_MANAGER

```
#define AGS_TYPE_LV2_URID_MANAGER                 (ags_lv2_urid_manager_get_type())
```

struct AgsLv2UridManager

```
struct AgsLv2UridManager;
```

struct AgsLv2UridManagerClass

```
struct AgsLv2UridManagerClass {
  GObjectClass gobject;
};
```

4.13 AgsLv2Worker

AgsLv2Worker — LV2 worker thread

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_WORKER_GET_OBJ_MUTEX()
#define	AGS_LV2_WORKER_RESPONSE_DATA()
gboolean	ags_lv2_worker_test_flags ()
void	ags_lv2_worker_set_flags ()
void	ags_lv2_worker_unset_flags ()
AgsLv2WorkerResponseData *	ags_lv2_worker_alloc_response_data ()
void	ags_lv2_worker_free_response_data ()
LV2_Worker_Status	ags_lv2_worker_respond ()
LV2_Worker_Status	ags_lv2_worker_schedule_work ()
void	ags_lv2_worker_do_poll ()
AgsLv2Worker *	ags_lv2_worker_new ()
#define	AGS_IS_LV2_WORKER()
#define	AGS_IS_LV2_WORKER_CLASS()
#define	AGS_LV2_WORKER()
#define	AGS_LV2_WORKER_CLASS()
#define	AGS_LV2_WORKER_GET_CLASS()
GType	ags_lv2_worker_flags_get_type ()
GType	ags_lv2_worker_get_type ()

Properties

gpointer	handle	Read / Write
gpointer	response-data	Read / Write
gpointer	work-data	Read / Write
guint	work-size	Read / Write
AgsWorkerThread *	worker-thread	Read / Write

Types and Values

enum	AgsLv2WorkerFlags
struct	AgsLv2WorkerResponseData
#define	AGS_TYPE_LV2_WORKER
struct	AgsLv2Worker
struct	AgsLv2WorkerClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsLv2WorkerFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsLv2Worker
```

Implemented Interfaces

AgsLv2Worker implements AgsConnectable.

Includes

```
#include <ags/plugin/ags_lv2_worker.h>
```

Description

The AgsLv2Worker acts as task queue thread.

Functions

AGS_LV2_WORKER_GET_OBJ_MUTEX()

```
#define AGS_LV2_WORKER_GET_OBJ_MUTEX(obj) (&(((AgsLv2Worker *) obj)->obj_mutex))
```

AGS_LV2_WORKER_RESPONSE_DATA()

```
#define AGS_LV2_WORKER_RESPONSE_DATA(ptr) ((AgsLv2WorkerResponseData *)(ptr))
```

ags_lv2_worker_test_flags ()

```
gboolean
ags_lv2_worker_test_flags (AgsLv2Worker *lv2_worker,
                           guint flags);
```

Test *flags* to be set on *recall* .

Parameters

lv2_worker	the AgsLv2Worker	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_lv2_worker_set_flags ()

```
void
ags_lv2_worker_set_flags (AgsLv2Worker *lv2_worker,
                          guint flags);
```

Set flags.

Parameters

lv2_worker	the AgsLv2Worker	
flags	the flags	

Since: 3.0.0

ags_lv2_worker_unset_flags ()

```
void
ags_lv2_worker_unset_flags (AgsLv2Worker *lv2_worker,
                            guint flags);
```

Unset flags.

Parameters

lv2_worker	the AgsLv2Worker	
flags	the flags	

Since: 3.0.0

ags_lv2_worker_alloc_response_data ()

```
AgsLv2WorkerResponseData~*
ags_lv2_worker_alloc_response_data ();
```

Allocate a new AgsLv2WorkerResponseData.

Returns

the new AgsLv2WorkerResponseData

Since: 3.0.0

ags_lv2_worker_free_response_data ()

```
void
ags_lv2_worker_free_response_data (AgsLv2WorkerResponseData *response_data);
```

Free *response_data* .

Parameters

response_data	the AgsLv2WorkerResponseData

Since: 3.0.0

ags_lv2_worker_respond ()

```
LV2_Worker_Status
ags_lv2_worker_respond (LV2_Worker_Respond_Handle handle,
                        uint32_t data_size,
                        const void *data);
```

Respond lv2 worker *handle* .

Parameters

handle	the LV2_Worker_Respond_Handle
data_size	the data size
data	the data

Returns

LV2_Worker_Status

Since: 3.0.0

ags_lv2_worker_schedule_work ()

```
LV2_Worker_Status
ags_lv2_worker_schedule_work (LV2_Worker_Schedule_Handle handle,
                              uint32_t data_size,
                              const void *data);
```

Schedule work of lv2 worker *handle* .

Parameters

handle	the LV2_Worker_Respond_Handle
data_size	the data size
data	the data

Returns

LV2_Worker_Status

Since: 3.0.0

ags_lv2_worker_do_poll ()

```
void
ags_lv2_worker_do_poll (AgsWorkerThread *worker_thread,
                        gpointer data);
```

Safe run callback.

Parameters

worker_thread	the AgsWorkerThread	
data	the data	

Since: 3.0.0

ags_lv2_worker_new ()

```
AgsLv2Worker~*
ags_lv2_worker_new ();
```

Create a new instance of AgsLv2Worker.

Returns

the new AgsLv2Worker

Since: 3.0.0

AGS_IS_LV2_WORKER()

```
#define AGS_IS_LV2_WORKER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_WORKER))
```

AGS_IS_LV2_WORKER_CLASS()

```
#define AGS_IS_LV2_WORKER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_WORKER))
```

AGS_LV2_WORKER()

```
#define AGS_LV2_WORKER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_WORKER, AgsLv2Worker))
```

AGS_LV2_WORKER_CLASS()

```
#define AGS_LV2_WORKER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_WORKER, AgsLv2WorkerClass))
```

AGS_LV2_WORKER_GET_CLASS()

```
#define AGS_LV2_WORKER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_WORKER, AgsLv2WorkerClass))
```

ags_lv2_worker_flags_get_type ()

```
GType
ags_lv2_worker_flags_get_type ();
```

ags_lv2_worker_get_type ()

```
GType
ags_lv2_worker_get_type (void);
```

Types and Values

enum AgsLv2WorkerFlags

Enum values to control the behavior or indicate internal state of AgsLv2Worker by enable/disable as flags.

Members

AGS_LV2_WORKER_RUN	the worker is running

struct AgsLv2WorkerResponseData

```
struct AgsLv2WorkerResponseData {
  uint32_t data_size;
  void *data;
};
```

AGS_TYPE_LV2_WORKER

```
#define AGS_TYPE_LV2_WORKER                 (ags_lv2_worker_get_type())
```

struct AgsLv2Worker

```
struct AgsLv2Worker;
```

struct AgsLv2WorkerClass

```
struct AgsLv2WorkerClass {
  GObjectClass gobject;
};
```

Property Details

The "handle" property

```
  "handle"                    gpointer
```

The assigned LV2_Handle

Owner: AgsLv2Worker

Flags: Read / Write

Since: 3.0.0

The "response-data" property

```
  "response-data"             gpointer
```

The assigned GList containing AgsLv2WorkerResponseData

Owner: AgsLv2Worker

Flags: Read / Write

Since: 3.0.0

The "work-data" property

```
  "work-data"                 gpointer
```

The assigned work data

Owner: AgsLv2Worker

Flags: Read / Write

Since: 3.0.0

The "work-size" property

```
  "work-size"                 guint
```

The work size.

Owner: AgsLv2Worker

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "worker-thread" property

```
"worker-thread"            AgsWorkerThread~*
```

The assigned worker-thread.

Owner: AgsLv2Worker

Flags: Read / Write

Since: 3.0.0

4.14 AgsLv2WorkerManager

AgsLv2WorkerManager — manage workers

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2_WORKER_MANAGER_GET_OBJ_MUTEX()
GObject *	ags_lv2_worker_manager_pull_worker ()
AgsLv2WorkerManager *	ags_lv2_worker_manager_get_instance ()
AgsLv2WorkerManager *	ags_lv2_worker_manager_new ()
#define	AGS_IS_LV2_WORKER_MANAGER()
#define	AGS_IS_LV2_WORKER_MANAGER_CLASS()
#define	AGS_LV2_WORKER_MANAGER()
#define	AGS_LV2_WORKER_MANAGER_CLASS()
#define	AGS_LV2_WORKER_MANAGER_GET_CLASS()
GType	ags_lv2_worker_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2_WORKER_MANAGER
struct	AgsLv2WorkerManager
struct	AgsLv2WorkerManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLv2WorkerManager
```

Includes

```
#include <ags/plugin/ags_lv2_worker_manager.h>
```

Description

The AgsLv2WorkerManager tracks your workers.

Functions

AGS_LV2_WORKER_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2_WORKER_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLv2WorkerManager *) obj)-> ↩
    obj_mutex))
```

ags_lv2_worker_manager_pull_worker ()

```
GObject~*
ags_lv2_worker_manager_pull_worker (AgsLv2WorkerManager *worker_manager);
```

Pull worker.

Parameters

worker_manager	the AgsLv2WorkerManager

Returns

the AgsWorkerThread.

[transfer full]

Since: 3.0.0

ags_lv2_worker_manager_get_instance ()

```
AgsLv2WorkerManager~*
ags_lv2_worker_manager_get_instance ();
```

Singleton function to optain the id manager instance.

Returns

an instance of AgsLv2WorkerManager.

[transfer none]

Since: 3.0.0

ags_lv2_worker_manager_new ()

```
AgsLv2WorkerManager~*
ags_lv2_worker_manager_new ();
```

Instantiate a id manager.

Returns

a new AgsLv2WorkerManager

Since: 3.0.0

AGS_IS_LV2_WORKER_MANAGER()

```
#define AGS_IS_LV2_WORKER_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2_WORKER_MANAGER))
```

AGS_IS_LV2_WORKER_MANAGER_CLASS()

```
#define AGS_IS_LV2_WORKER_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2_WORKER_MANAGER))
```

AGS_LV2_WORKER_MANAGER()

```
#define AGS_LV2_WORKER_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2_WORKER_MANAGER, AgsLv2WorkerManager))
```

AGS_LV2_WORKER_MANAGER_CLASS()

```
#define AGS_LV2_WORKER_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2_WORKER_MANAGER, AgsLv2WorkerManagerClass))
```

AGS_LV2_WORKER_MANAGER_GET_CLASS()

```
#define AGS_LV2_WORKER_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2_WORKER_MANAGER, AgsLv2WorkerManagerClass))
```

ags_lv2_worker_manager_get_type ()

```
GType
ags_lv2_worker_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2_WORKER_MANAGER

```
#define AGS_TYPE_LV2_WORKER_MANAGER                 (ags_lv2_worker_manager_get_type())
```

struct AgsLv2WorkerManager

```
struct AgsLv2WorkerManager;
```

struct AgsLv2WorkerManagerClass

```
struct AgsLv2WorkerManagerClass {
  GObjectClass gobject;
};
```

4.15 AgsLv2uiManager

AgsLv2uiManager — Singleton pattern to organize LV2UI

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2UI_MANAGER_GET_OBJ_MUTEX()
gchar **	ags_lv2ui_manager_get_default_path ()
void	ags_lv2ui_manager_set_default_path ()
gchar **	ags_lv2ui_manager_get_filenames ()
AgsLv2uiPlugin *	ags_lv2ui_manager_find_lv2ui_plugin ()
AgsLv2uiPlugin *	ags_lv2ui_manager_find_lv2ui_plugin_with_index ()
void	ags_lv2ui_manager_load_blacklist ()
void	ags_lv2ui_manager_load_file ()
void	ags_lv2ui_manager_load_default_directory ()
AgsLv2uiManager *	ags_lv2ui_manager_get_instance ()
AgsLv2uiManager *	ags_lv2ui_manager_new ()
#define	AGS_IS_LV2UI_MANAGER()
#define	AGS_IS_LV2UI_MANAGER_CLASS()
#define	AGS_LV2UI_MANAGER()
#define	AGS_LV2UI_MANAGER_CLASS()
#define	AGS_LV2UI_MANAGER_GET_CLASS()
GType	ags_lv2ui_manager_get_type ()

Types and Values

#define	AGS_TYPE_LV2UI_MANAGER
struct	AgsLv2uiManager
struct	AgsLv2uiManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLv2uiManager
```

Includes

```
#include <ags/plugin/ags_lv2ui_manager.h>
```

Description

The AgsLv2uiManager loads/unloads LV2UI plugins.

Functions

AGS_LV2UI_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_LV2UI_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsLv2uiManager *) obj)->obj_mutex))
```

ags_lv2ui_manager_get_default_path ()

```
gchar~**
ags_lv2ui_manager_get_default_path ();
```

Get lv2ui manager default plugin path.

Returns

the plugin default search path as a string vector.

[element-type utf8][array zero-terminated=1][transfer none]

Since: 3.0.0

ags_lv2ui_manager_set_default_path ()

```
void
ags_lv2ui_manager_set_default_path (gchar **default_path);
```

Set lv2ui manager default plugin path.

Parameters

default_path	the string vector array to use as default path.	*[element-type utf8][array zero-terminated=1][transfer full]*

Since: 3.0.0

ags_lv2ui_manager_get_filenames ()

```
gchar~**
ags_lv2ui_manager_get_filenames (AgsLv2uiManager *lv2ui_manager);
```

Retrieve all filenames

Parameters

lv2ui_manager	the AgsLv2uiManager	

Returns

a NULL-terminated array of filenames.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_lv2ui_manager_find_lv2ui_plugin ()

```
AgsLv2uiPlugin~*
ags_lv2ui_manager_find_lv2ui_plugin (AgsLv2uiManager *lv2ui_manager,
                                     gchar *ui_filename,
                                     gchar *ui_effect);
```

Lookup UI filename in loaded plugins.

Parameters

lv2ui_manager	the AgsLv2uiManager	
ui_filename	the UI filename of the plugin	
ui_effect	the UI effect's name	

Returns

the AgsLv2uiPlugin.

[transfer none]

Since: 3.0.0

ags_lv2ui_manager_find_lv2ui_plugin_with_index ()

```
AgsLv2uiPlugin~*
ags_lv2ui_manager_find_lv2ui_plugin_with_index
                                (AgsLv2uiManager *lv2ui_manager,
                                 gchar *ui_filename,
                                 guint ui_effect_index);
```

Lookup *ui_filename* with *ui_effect_index* in loaded plugins.

Parameters

lv2ui_manager	the AgsLv2uiManager	
ui_filename	the UI filename of the plugin	
ui_effect_index	the UI index	

Returns

the AgsLv2uiPlugin.

[transfer none]

Since: 3.0.0

ags_lv2ui_manager_load_blacklist ()

```
void
ags_lv2ui_manager_load_blacklist (AgsLv2uiManager *lv2ui_manager,
                                  gchar *blacklist_filename);
```

Load blacklisted plugin filenames.

Parameters

lv2ui_manager	the AgsLv2uiManager	
blacklist_filename	the filename as string	

Since: 3.0.0

ags_lv2ui_manager_load_file ()

```
void
ags_lv2ui_manager_load_file (AgsLv2uiManager *lv2ui_manager,
                             AgsTurtle *manifest,
                             AgsTurtle *turtle,
                             gchar *lv2ui_path,
                             gchar *filename);
```

Load *filename* specified plugin.

Parameters

lv2ui_manager	the AgsLv2uiManager	
manifest	the manifest	
turtle	the AgsTurtle	
lv2ui_path	the lv2ui path	
filename	the filename of the plugin	

Since: 3.0.0

ags_lv2ui_manager_load_default_directory ()

```
void
ags_lv2ui_manager_load_default_directory
                                (AgsLv2uiManager *lv2ui_manager);
```

Loads all available plugins.

Parameters

lv2ui_manager	the AgsLv2uiManager	

Since: 3.0.0

ags_lv2ui_manager_get_instance ()

```
AgsLv2uiManager~*
ags_lv2ui_manager_get_instance ();
```

Get instance.

Returns

the AgsLv2uiManager.

[transfer none]

Since: 3.0.0

ags_lv2ui_manager_new ()

```
AgsLv2uiManager~*
ags_lv2ui_manager_new ();
```

Creates an AgsLv2uiManager

Returns

a new AgsLv2uiManager

Since: 3.0.0

AGS_IS_LV2UI_MANAGER()

```
#define AGS_IS_LV2UI_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2UI_MANAGER))
```

AGS_IS_LV2UI_MANAGER_CLASS()

```
#define AGS_IS_LV2UI_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2UI_MANAGER))
```

AGS_LV2UI_MANAGER()

```
#define AGS_LV2UI_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2UI_MANAGER, AgsLv2uiManager))
```

AGS_LV2UI_MANAGER_CLASS()

```
#define AGS_LV2UI_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2UI_MANAGER, AgsLv2uiManagerClass))
```

AGS_LV2UI_MANAGER_GET_CLASS()

```
#define AGS_LV2UI_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2UI_MANAGER, AgsLv2uiManagerClass))
```

ags_lv2ui_manager_get_type ()

```
GType
ags_lv2ui_manager_get_type (void);
```

Types and Values

AGS_TYPE_LV2UI_MANAGER

```
#define AGS_TYPE_LV2UI_MANAGER                (ags_lv2ui_manager_get_type())
```

struct AgsLv2uiManager

```
struct AgsLv2uiManager;
```

struct AgsLv2uiManagerClass

```
struct AgsLv2uiManagerClass {
  GObjectClass gobject;
};
```

4.16 AgsLv2uiPlugin

AgsLv2uiPlugin — The lv2ui plugin class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LV2UI_PLUGIN_DESCRIPTOR()
gboolean	ags_lv2ui_plugin_test_flags ()
void	ags_lv2ui_plugin_set_flags ()
void	ags_lv2ui_plugin_unset_flags ()
GList *	ags_lv2ui_plugin_find_gui_uri ()
AgsLv2uiPlugin *	ags_lv2ui_plugin_new ()
#define	AGS_IS_LV2UI_PLUGIN()
#define	AGS_IS_LV2UI_PLUGIN_CLASS()
#define	AGS_LV2UI_PLUGIN()
#define	AGS_LV2UI_PLUGIN_CLASS()
#define	AGS_LV2UI_PLUGIN_GET_CLASS()
GType	ags_lv2ui_plugin_flags_get_type ()
GType	ags_lv2ui_plugin_get_type ()

Properties

AgsTurtle *	gui-turtle	Read / Write
char *	gui-uri	Read / Write
AgsLv2Plugin *	lv2-plugin	Read / Write
AgsTurtle *	manifest	Read / Write

Types and Values

enum	AgsLv2uiPluginFlags
#define	AGS_TYPE_LV2UI_PLUGIN
struct	AgsLv2uiPlugin
struct	AgsLv2uiPluginClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsLv2uiPluginFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsBasePlugin
        &#x2570;&#x2500;&#x2500; AgsLv2uiPlugin
```

Includes

```
#include <ags/plugin/ags_lv2ui_plugin.h>
```

Description

The AgsLv2uiPlugin loads/unloads a Lv2ui plugin.

Functions

AGS_LV2UI_PLUGIN_DESCRIPTOR()

```
#define AGS_LV2UI_PLUGIN_DESCRIPTOR(ptr) ((LV2UI_Descriptor *)(ptr))
```

ags_lv2ui_plugin_test_flags ()

```
gboolean
ags_lv2ui_plugin_test_flags (AgsLv2uiPlugin *lv2ui_plugin,
                             guint flags);
```

Test *flags* to be set on *recall* .

Parameters

lv2ui_plugin	the AgsLv2uiPlugin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_lv2ui_plugin_set_flags ()

```
void
ags_lv2ui_plugin_set_flags (AgsLv2uiPlugin *lv2ui_plugin,
                            guint flags);
```

Set flags.

Parameters

lv2ui_plugin	the AgsLv2uiPlugin	
flags	the flags	

Since: 3.0.0

ags_lv2ui_plugin_unset_flags ()

```
void
ags_lv2ui_plugin_unset_flags (AgsLv2uiPlugin *lv2ui_plugin,
                              guint flags);
```

Unset flags.

Parameters

lv2ui_plugin	the AgsLv2uiPlugin	
flags	the flags	

Since: 3.0.0

ags_lv2ui_plugin_find_gui_uri ()

```
GList~*
ags_lv2ui_plugin_find_gui_uri (GList *lv2ui_plugin,
                               gchar *gui_uri);
```

Find next matching gui-uri in *lv2ui_plugin* .

Parameters

lv2ui_plugin	the GList containing AgsLv2uiPlugin.	*[element-type AgsAudio.Lv2uiPlugin][transfer none]*
gui_uri	the gui-uri as string	

Returns

the next matching GList.

[element-type AgsAudio.Lv2uiPlugin][transfer none]

Since: 3.0.0

ags_lv2ui_plugin_new ()

```
AgsLv2uiPlugin~*
ags_lv2ui_plugin_new (AgsTurtle *gui_turtle,
                      gchar *filename,
                      gchar *effect,
                      gchar *gui_uri,
                      guint effect_index);
```

Creates an AgsLv2uiPlugin

Parameters

gui_turtle	the AgsTurtle	
filename	the plugin .so	
effect	the effect's string representation	
gui_uri	the effect's gui_uri	
effect_index	the effect's index	

Returns

a new AgsLv2uiPlugin

Since: 3.0.0

AGS_IS_LV2UI_PLUGIN()

```
#define AGS_IS_LV2UI_PLUGIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_LV2UI_PLUGIN))
```

AGS_IS_LV2UI_PLUGIN_CLASS()

```
#define AGS_IS_LV2UI_PLUGIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_LV2UI_PLUGIN))
```

AGS_LV2UI_PLUGIN()

```
#define AGS_LV2UI_PLUGIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LV2UI_PLUGIN, AgsLv2uiPlugin))
```

AGS_LV2UI_PLUGIN_CLASS()

```
#define AGS_LV2UI_PLUGIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LV2UI_PLUGIN, AgsLv2uiPluginClass))
```

AGS_LV2UI_PLUGIN_GET_CLASS()

```
#define AGS_LV2UI_PLUGIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_LV2UI_PLUGIN, AgsLv2uiPluginClass))
```

ags_lv2ui_plugin_flags_get_type ()

```
GType
ags_lv2ui_plugin_flags_get_type ();
```

ags_lv2ui_plugin_get_type ()

```
GType
ags_lv2ui_plugin_get_type (void);
```

Types and Values

enum AgsLv2uiPluginFlags

Enum values to control the behavior or indicate internal state of AgsLv2uiPlugin by enable/disable as flags.

Members

AGS_LV2UI_PLUGIN_IS_SYNTHESIZER	is synthesizer
AGS_LV2UI_PLUGIN_GTK2	has Gtk+-2.0 UI
AGS_LV2UI_PLUGIN_GTK3	has Gtk3 UI
AGS_LV2UI_PLUGIN_QT4	has Qt4 UI
AGS_LV2UI_PLUGIN_QT5	has Qt5 UI

AGS_TYPE_LV2UI_PLUGIN

```
#define AGS_TYPE_LV2UI_PLUGIN                (ags_lv2ui_plugin_get_type())
```

struct AgsLv2uiPlugin

```
struct AgsLv2uiPlugin;
```

struct AgsLv2uiPluginClass

```
struct AgsLv2uiPluginClass {
  AgsBasePluginClass base_plugin;
};
```

Property Details

The "gui-turtle" property

```
"gui-turtle"                AgsTurtle~*
```

The assigned GUI turtle.

Owner: AgsLv2uiPlugin

Flags: Read / Write

Since: 3.0.0

The "gui-uri" property

```
"gui-uri"                   char~*
```

The assigned GUI URI.

Owner: AgsLv2uiPlugin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "lv2-plugin" property

```
"lv2-plugin"                AgsLv2Plugin~*
```

The assigned LV2 plugin.

Owner: AgsLv2uiPlugin

Flags: Read / Write

Since: 3.0.0

The "manifest" property

```
"manifest"                  AgsTurtle~*
```

The assigned manifest.

Owner: AgsLv2uiPlugin

Flags: Read / Write

Since: 3.0.0

Part II

The audio layer

The audio layer contains various objects related to audio processing.

Chapter 5

Utility functions

The utility functions do common tasks.

5.1 AgsSoundEnums

AgsSoundEnums — sound enumerations

Stability Level

Stable, unless otherwise indicated

Functions

GType	ags_sound_ability_flags_get_type ()
GType	ags_sound_behaviour_flags_get_type ()
GType	ags_sound_key_format_get_type ()
GType	ags_sound_scope_get_type ()
GType	ags_sound_staging_flags_get_type ()
GType	ags_sound_state_flags_get_type ()

Types and Values

enum	AgsSoundAbilityFlags
enum	AgsSoundBehaviourFlags
enum	AgsSoundScope
enum	AgsSoundStagingFlags
enum	AgsSoundStateFlags
enum	AgsSoundKeyFormat
#define	AGS_TYPE_SOUND_ABILITY_FLAGS
#define	AGS_TYPE_SOUND_BEHAVIOUR_FLAGS
#define	AGS_TYPE_SOUND_KEY_FORMAT
#define	AGS_TYPE_SOUND_SCOPE
#define	AGS_TYPE_SOUND_STAGING_FLAGS
#define	AGS_TYPE_SOUND_STATE_FLAGS

Object Hierarchy

```
    GEnum
    &#x251c;&#x2500;&#x2500; AgsSoundKeyFormat
    &#x2570;&#x2500;&#x2500; AgsSoundScope
    GFlags
    &#x251c;&#x2500;&#x2500; AgsSoundAbilityFlags
    &#x251c;&#x2500;&#x2500; AgsSoundBehaviourFlags
    &#x251c;&#x2500;&#x2500; AgsSoundStagingFlags
    &#x2570;&#x2500;&#x2500; AgsSoundStateFlags
```

Includes

```
#include <ags/audio/ags_sound_enums.h>
```

Description

Enumerations related to sound.

Functions

ags_sound_ability_flags_get_type ()

```
GType
ags_sound_ability_flags_get_type ();
```

ags_sound_behaviour_flags_get_type ()

```
GType
ags_sound_behaviour_flags_get_type ();
```

ags_sound_key_format_get_type ()

```
GType
ags_sound_key_format_get_type ();
```

ags_sound_scope_get_type ()

```
GType
ags_sound_scope_get_type ();
```

ags_sound_staging_flags_get_type ()

```
GType
ags_sound_staging_flags_get_type ();
```

ags_sound_state_flags_get_type ()

```
GType
ags_sound_state_flags_get_type ();
```

Types and Values

enum AgsSoundAbilityFlags

Enum values to specify particular ability. Mark an object to be able to handle matching scope.

Members

AGS_SOUND_ABILITY_PLAYBACK	playback abil- ity
AGS_SOUND_ABILITY_SEQUENCER	sequenecer abil- ity
AGS_SOUND_ABILITY_NOTATION	notation abil- i- tiy
AGS_SOUND_ABILITY_WAVE	wave abil- i- tiy
AGS_SOUND_ABILITY_MIDI	MIDI abil- ity

enum AgsSoundBehaviourFlags

Enum values to describe behaviour.

Members

AGS_SOUND_BEHAVIOUR_PATTERN_MODE	pattern mode
AGS_SOUND_BEHAVIOUR_BULK_MODE	bulk mode
AGS_SOUND_BEHAVIOUR_REVERSE_MAPPING	reverse map- ping
AGS_SOUND_BEHAVIOUR_DEFAULTS_TO_OUTPUT	defaults to out- put
AGS_SOUND_BEHAVIOUR_DEFAULTS_TO_INPUT	defaults to in- put

AGS_SOUND_BEHAVIOUR_CHAINED_TO_OUTPUT	chained to output
AGS_SOUND_BEHAVIOUR_CHAINED_TO_INPUT	chained to input
AGS_SOUND_BEHAVIOUR_PERSISTENT	persistent
AGS_SOUND_BEHAVIOUR_PERSISTENT_PLAYBACK	persistent playback
AGS_SOUND_BEHAVIOUR_PERSISTENT_NOTATION	persistent notation
AGS_SOUND_BEHAVIOUR_PERSISTENT_SEQUENCER	persistent sequencer
AGS_SOUND_BEHAVIOUR_PERSISTENT_WAVE	persistent wave
AGS_SOUND_BEHAVIOUR_PERSISTENT_MIDI	persistent MIDI
AGS_SOUND_BEHAVIOUR_PROPAGATE_DONE	propagate done

enum AgsSoundScope

Enum values to specify current scope or request current scope.

Members

AGS_SOUND_SCOPE_PLAYBACK	playback scope
AGS_SOUND_SCOPE_SEQUENCER	sequencer scope
AGS_SOUND_SCOPE_NOTATION	notation ability
AGS_SOUND_SCOPE_WAVE	wave ability
AGS_SOUND_SCOPE_MIDI	MIDI scope
AGS_SOUND_SCOPE_LAST	the last scope

enum AgsSoundStagingFlags

Enum values to stage the specified action.

Members

AGS_SOUND_STAGING_CHECK_RT_DATA	check rt-data
AGS_SOUND_STAGING_RUN_INIT_PRE	run init pre
AGS_SOUND_STAGING_RUN_INIT_INTER	run init in-ter
AGS_SOUND_STAGING_RUN_INIT_POST	run init post
AGS_SOUND_STAGING_FEED_INPUT_QUEUE	feed in-put queue
AGS_SOUND_STAGING_AUTOMATE	automate
AGS_SOUND_STAGING_RUN_PRE	run pre
AGS_SOUND_STAGING_RUN_INTER	run in-ter
AGS_SOUND_STAGING_RUN_POST	run post
AGS_SOUND_STAGING_DO_FEEDBACK	do feed-back
AGS_SOUND_STAGING_FEED_OUTPUT_QUEUE	feed out-put queue
AGS_SOUND_STAGING_FINI	fini
AGS_SOUND_STAGING_CANCEL	cancel
AGS_SOUND_STAGING_DONE	done
AGS_SOUND_STAGING_REMOVE	remove
AGS_SOUND_STAGING_RESET	reset
AGS_SOUND_STAGING_FX	fx stag-ing

enum AgsSoundStateFlags

Enum values to indicate state.

Members

AGS_SOUND_STATE_IS_WAITING	is wait-ing
AGS_SOUND_STATE_IS_ACTIVE	is ac-tive

AGS_SOUND_STATE_IS_PROCESSING	is processing
AGS_SOUND_STATE_IS_TERMINATING	is terminating

enum AgsSoundKeyFormat

Enum values to specify key format.

Members

AGS_SOUND_KEY_FORMAT_16TH	key format 16th
AGS_SOUND_KEY_FORMAT_256TH	key format 256th

AGS_TYPE_SOUND_ABILITY_FLAGS

```
#define AGS_TYPE_SOUND_ABILITY_FLAGS        (ags_sound_ability_flags_get_type())
```

AGS_TYPE_SOUND_BEHAVIOUR_FLAGS

```
#define AGS_TYPE_SOUND_BEHAVIOUR_FLAGS      (ags_sound_behaviour_flags_get_type())
```

AGS_TYPE_SOUND_KEY_FORMAT

```
#define AGS_TYPE_SOUND_KEY_FORMAT           (ags_sound_key_format_get_type())
```

AGS_TYPE_SOUND_SCOPE

```
#define AGS_TYPE_SOUND_SCOPE                (ags_sound_scope_get_type())
```

AGS_TYPE_SOUND_STAGING_FLAGS

```
#define AGS_TYPE_SOUND_STAGING_FLAGS        (ags_sound_staging_flags_get_type())
```

AGS_TYPE_SOUND_STATE_FLAGS

```
#define AGS_TYPE_SOUND_STATE_FLAGS          (ags_sound_state_flags_get_type())
```

5.2 AgsSynthEnums

AgsSynthEnums — synth enumerations

Stability Level

Stable, unless otherwise indicated

Functions

GType	ags_synth_oscillator_mode_get_type ()

Types and Values

enum	AgsSynthOscillatorMode
#define	AGS_TYPE_SYNTH_OSCILLATOR_MODE

Object Hierarchy

```
    GEnum
    &#x2570;&#x2500;&#x2500; AgsSynthOscillatorMode
```

Includes

```
#include <ags/audio/ags_synth_enums.h>
```

Description

Enumerations related to synth.

Functions

ags_synth_oscillator_mode_get_type ()

```
GType
ags_synth_oscillator_mode_get_type ();
```

Types and Values

enum AgsSynthOscillatorMode

Enum values to specify oscillator mode.

Members

AGS_SYNTH_OSCILLATOR_SIN	sine os-cil-la-tor
AGS_SYNTH_OSCILLATOR_SAWTOOTH	sawtooth os-cil-la-tor
AGS_SYNTH_OSCILLATOR_TRIANGLE	triangle os-cil-la-tor
AGS_SYNTH_OSCILLATOR_SQUARE	square os-cil-la-tor
AGS_SYNTH_OSCILLATOR_IMPULSE	impulse os-cil-la-tor
AGS_SYNTH_OSCILLATOR_LAST	last mode

AGS_TYPE_SYNTH_OSCILLATOR_MODE

```
#define AGS_TYPE_SYNTH_OSCILLATOR_MODE                (ags_synth_oscillator_mode_get_type())
```

5.3 AgsSoundcardUtil

AgsSoundcardUtil — soundcard util

Stability Level

Stable, unless otherwise indicated

Functions

GRecMutex *	ags_soundcard_util_get_obj_mutex ()
void	ags_soundcard_util_adjust_delay_and_attack ()
GType	ags_soundcard_util_get_type ()

Types and Values

#define	AGS_TYPE_SOUNDCARD_UTIL
struct	AgsSoundcardUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsSoundcardUtil
```

Includes

```
#include <ags/audio/ags_soundcard_util.h>
```

Description

Soundcard utility functions.

Functions

ags_soundcard_util_get_obj_mutex ()

```
GRecMutex~*
ags_soundcard_util_get_obj_mutex (GObject *soundcard);
```

Get object mutex of *soundcard* .

Parameters

soundcard	the GObject sub-type implementing AgsSoundcard

Returns

GRecMutex pointer.

[transfer none]

Since: 3.0.0

ags_soundcard_util_adjust_delay_and_attack ()

```
void
ags_soundcard_util_adjust_delay_and_attack
                                (GObject *soundcard);
```

Adjust delay and attack of *soundcard* .

Parameters

soundcard	the GObject sub-type implementing AgsSoundcard

Since: 3.3.0

ags_soundcard_util_get_type ()

```
GType
ags_soundcard_util_get_type (void);
```

Types and Values

AGS_TYPE_SOUNDCARD_UTIL

```
#define AGS_TYPE_SOUNDCARD_UTIL         (ags_soundcard_util_get_type())
```

struct AgsSoundcardUtil

```
struct AgsSoundcardUtil {
  //empty
};
```

5.4 AgsSequencerUtil

AgsSequencerUtil — sequencer util

Stability Level

Stable, unless otherwise indicated

Functions

GRecMutex *	ags_sequencer_util_get_obj_mutex ()
GType	ags_sequencer_util_get_type ()

Types and Values

#define	AGS_TYPE_SEQUENCER_UTIL
struct	AgsSequencerUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsSequencerUtil
```

Includes

```
#include <ags/audio/ags_sequencer_util.h>
```

Description

Sequencer utility functions.

Functions

ags_sequencer_util_get_obj_mutex ()

```
GRecMutex~*
ags_sequencer_util_get_obj_mutex (GObject *sequencer);
```

Get object mutex of *sequencer* .

Parameters

sequencer	the GObject sub-type implementing AgsSoundcard

Returns

GRecMutex pointer.

[transfer none]

Since: 3.0.0

ags_sequencer_util_get_type ()

```
GType
ags_sequencer_util_get_type (void);
```

Types and Values

AGS_TYPE_SEQUENCER_UTIL

```
#define AGS_TYPE_SEQUENCER_UTIL         (ags_sequencer_util_get_type())
```

struct AgsSequencerUtil

```
struct AgsSequencerUtil {
  //empty
};
```

5.5 AgsCharBufferUtil

AgsCharBufferUtil — util functions to handle char buffer

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_char_buffer_util_copy_s8_to_cbuffer ()
void	ags_char_buffer_util_copy_s16_to_cbuffer ()
void	ags_char_buffer_util_copy_s24_to_cbuffer ()
void	ags_char_buffer_util_copy_s32_to_cbuffer ()
void	ags_char_buffer_util_copy_s64_to_cbuffer ()
void	ags_char_buffer_util_copy_float_to_cbuffer ()
void	ags_char_buffer_util_copy_double_to_cbuffer ()
void	ags_char_buffer_util_copy_cbuffer_to_s8 ()
void	ags_char_buffer_util_copy_cbuffer_to_s16 ()
void	ags_char_buffer_util_copy_cbuffer_to_s24 ()
void	ags_char_buffer_util_copy_cbuffer_to_s32 ()
void	ags_char_buffer_util_copy_cbuffer_to_s64 ()
void	ags_char_buffer_util_copy_cbuffer_to_float ()
void	ags_char_buffer_util_copy_cbuffer_to_double ()
void	ags_char_buffer_util_copy_buffer_to_buffer ()
GType	ags_char_buffer_util_get_type ()

Types and Values

enum	AgsCharBufferUtilCopyMode
#define	AGS_TYPE_CHAR_BUFFER_UTIL
struct	AgsCharBufferUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsCharBufferUtil
```

Includes

```
#include <ags/audio/ags_char_buffer_util.h>
```

Description

These utility functions allow you to store/restore PCM data from char buffer.

Functions

ags_char_buffer_util_copy_s8_to_cbuffer ()

```
void
ags_char_buffer_util_copy_s8_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gint8 *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy gint8 to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_s16_to_cbuffer ()

```
void
ags_char_buffer_util_copy_s16_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gint16 *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy gint16 to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_s24_to_cbuffer ()

```
void
ags_char_buffer_util_copy_s24_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy gint32 to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_s32_to_cbuffer ()

```
void
ags_char_buffer_util_copy_s32_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy gint32 to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_s64_to_cbuffer ()

```
void
ags_char_buffer_util_copy_s64_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gint64 *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy gint64 to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_float_to_cbuffer ()

```
void
ags_char_buffer_util_copy_float_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy float to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_double_to_cbuffer ()

```
void
ags_char_buffer_util_copy_double_to_cbuffer
                                (guchar *destination,
                                 guint word_size,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy double to char buffer.

Parameters

destination	destination buffer	
word_size	word size	
dchannels	destination channels	
source	source buffer	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_s8 ()

```
void
ags_char_buffer_util_copy_cbuffer_to_s8
                                (gint8 *destination,
                                 guint dchannels,
                                 guchar *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to gint8 buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_s16 ()

```
void
ags_char_buffer_util_copy_cbuffer_to_s16
                                (gint16 *destination,
                                 guint dchannels,
                                 guchar *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to gint16 buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_s24 ()

```
void
ags_char_buffer_util_copy_cbuffer_to_s24
                                (gint32 *destination,
                                 guint dchannels,
                                 guchar *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to gint32 buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_s32 ()

```
void
ags_char_buffer_util_copy_cbuffer_to_s32
                                (gint32 *destination,
                                 guint dchannels,
                                 guchar *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to gint32 buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_s64 ()

```
void
ags_char_buffer_util_copy_cbuffer_to_s64
                                (gint64 *destination,
                                 guint dchannels,
                                 guchar *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to gint64 buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_float ()

```
void
ags_char_buffer_util_copy_cbuffer_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gint8 *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to float buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_cbuffer_to_double ()

```
void
ags_char_buffer_util_copy_cbuffer_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 guchar *source,
                                 guint word_size,
                                 guint schannels,
                                 guint frame_count,
                                 guint byte_order);
```

Copy char to double buffer.

Parameters

destination	destination buffer	
dchannels	destination channels	
source	source buffer	
word_size	word size	
schannels	source channels	
frame_count	frame count	
byte_order	the byte-order	

Since: 3.0.0

ags_char_buffer_util_copy_buffer_to_buffer ()

```
void
ags_char_buffer_util_copy_buffer_to_buffer
                                (void *destination,
                                 guint dchannels,
                                 guint doffset,
                                 void *source,
                                 guint schannels,
                                 guint soffset,
                                 guint frame_count,
                                 guint byte_order,
                                 guint word_size,
                                 guint mode);
```

ags_char_buffer_util_get_type ()

```
GType
ags_char_buffer_util_get_type (void);
```

Types and Values

enum AgsCharBufferUtilCopyMode

Copy modes.

Members

AGS_CHAR_BUFFER_UTIL_COPY_S8_TO_CBUFFER	copy signed 8 bit to char buffer
AGS_CHAR_BUFFER_UTIL_COPY_S16_TO_CBUFFER	copy signed 16 bit to char buffer
AGS_CHAR_BUFFER_UTIL_COPY_S24_TO_CBUFFER	copy signed 24 bit to char buffer

AGS_CHAR_BUFFER_UTIL_COPY_S32_TO_CBUFFER	copy signed 32 bit to char buffer
AGS_CHAR_BUFFER_UTIL_COPY_S64_TO_CBUFFER	copy signed 64 bit to char buffer
AGS_CHAR_BUFFER_UTIL_COPY_FLOAT_TO_CBUFFER	copy float to char buffer
AGS_CHAR_BUFFER_UTIL_COPY_DOUBLE_TO_CBUFFER	copy dou- ble to char buffer
AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_S8	copy char buffer to signed 8 bit
AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_S16	copy char buffer to signed 16 bit
AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_S24	copy char buffer to signed 24 bit
AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_S32	copy char buffer to signed 32 bit

AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_S64	copy char buffer to signed 64 bit
AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_FLOAT	copy char buffer to float
AGS_CHAR_BUFFER_UTIL_COPY_CBUFFER_TO_DOUBLE	copy char buffer to dou- ble

AGS_TYPE_CHAR_BUFFER_UTIL

```
#define AGS_TYPE_CHAR_BUFFER_UTIL          (ags_char_buffer_util_get_type())
```

struct AgsCharBufferUtil

```
struct AgsCharBufferUtil {
  //empty
};
```

5.6 AgsChorusUtil

AgsChorusUtil — chorus util

Stability Level

Stable, unless otherwise indicated

Functions

AgsChorusUtil *	ags_chorus_util_alloc ()
gpointer	ags_chorus_util_copy ()
void	ags_chorus_util_free ()
gpointer	ags_chorus_util_get_source ()
void	ags_chorus_util_set_source ()
guint	ags_chorus_util_get_source_stride ()
void	ags_chorus_util_set_source_stride ()
gpointer	ags_chorus_util_get_destination ()
void	ags_chorus_util_set_destination ()
guint	ags_chorus_util_get_destination_stride ()
void	ags_chorus_util_set_destination_stride ()

guint	ags_chorus_util_get_buffer_length ()
void	ags_chorus_util_set_buffer_length ()
guint	ags_chorus_util_get_format ()
void	ags_chorus_util_set_format ()
guint	ags_chorus_util_get_samplerate ()
void	ags_chorus_util_set_samplerate ()
guint64	ags_chorus_util_get_offset ()
void	ags_chorus_util_set_offset ()
gdouble	ags_chorus_util_get_base_key ()
void	ags_chorus_util_set_base_key ()
gdouble	ags_chorus_util_get_input_volume ()
void	ags_chorus_util_set_input_volume ()
gdouble	ags_chorus_util_get_output_volume ()
void	ags_chorus_util_set_output_volume ()
guint	ags_chorus_util_get_lfo_oscillator ()
void	ags_chorus_util_set_lfo_oscillator ()
gdouble	ags_chorus_util_get_lfo_frequency ()
void	ags_chorus_util_set_lfo_frequency ()
gdouble	ags_chorus_util_get_depth ()
void	ags_chorus_util_set_depth ()
gdouble	ags_chorus_util_get_mix ()
void	ags_chorus_util_set_mix ()
gdouble	ags_chorus_util_get_delay ()
void	ags_chorus_util_set_delay ()
GType	ags_chorus_util_get_pitch_type ()
void	ags_chorus_util_set_pitch_type ()
gpointer	ags_chorus_util_get_pitch_util ()
void	ags_chorus_util_set_pitch_util ()
void	ags_chorus_util_compute_s8 ()
void	ags_chorus_util_compute_s16 ()
void	ags_chorus_util_compute_s24 ()
void	ags_chorus_util_compute_s32 ()
void	ags_chorus_util_compute_s64 ()
void	ags_chorus_util_compute_float ()
void	ags_chorus_util_compute_double ()
void	ags_chorus_util_compute_complex ()
void	ags_chorus_util_compute ()
#define	AGS_CHORUS_UTIL()
GType	ags_chorus_util_get_type ()

Types and Values

#define	AGS_CHORUS_UTIL_DEFAULT_PITCH_MIX_BUFFER_SIZE
#define	AGS_CHORUS_UTIL_DEFAULT_HISTORY_BUFFER_LENGTH
#define	AGS_CHORUS_UTIL_DEFAULT_LFO_OSCILLATOR
#define	AGS_CHORUS_UTIL_DEFAULT_LFO_FREQUENCY
#define	AGS_CHORUS_UTIL_DEFAULT_DEPTH
#define	AGS_CHORUS_UTIL_DEFAULT_MIX
#define	AGS_CHORUS_UTIL_DEFAULT_DELAY
#define	AGS_TYPE_CHORUS_UTIL
struct	AgsChorusUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsChorusUtil
```

Includes

```
#include <ags/audio/ags_chorus_util.h>
```

Description

Utility functions to chorus.

Functions

ags_chorus_util_alloc ()

```
AgsChorusUtil~*
ags_chorus_util_alloc ();
```

Allocate AgsChorusUtil.

Returns

the newly allocated AgsChorusUtil

Since: 3.13.4

ags_chorus_util_copy ()

```
gpointer
ags_chorus_util_copy (AgsChorusUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsChorusUtil

Returns

a pointer of the new AgsChorusUtil

Since: 3.13.4

ags_chorus_util_free ()

```
void
ags_chorus_util_free (AgsChorusUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_get_source ()

```
gpointer
ags_chorus_util_get_source (AgsChorusUtil *chorus_util);
```

Get source buffer of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the source buffer

Since: 3.13.4

ags_chorus_util_set_source ()

```
void
ags_chorus_util_set_source (AgsChorusUtil *chorus_util,
                            gpointer source);
```

Set *source* buffer of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil
source	the source buffer

Since: 3.13.4

ags_chorus_util_get_source_stride ()

```
guint
ags_chorus_util_get_source_stride (AgsChorusUtil *chorus_util);
```

Get source stride of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the source buffer stride

Since: 3.13.4

ags_chorus_util_set_source_stride ()

```
void
ags_chorus_util_set_source_stride (AgsChorusUtil *chorus_util,
                                   guint source_stride);
```

Set *source* stride of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
source_stride	the source buffer stride	

Since: 3.13.4

ags_chorus_util_get_destination ()

```
gpointer
ags_chorus_util_get_destination (AgsChorusUtil *chorus_util);
```

Get destination buffer of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the destination buffer

Since: 3.13.4

ags_chorus_util_set_destination ()

```
void
ags_chorus_util_set_destination (AgsChorusUtil *chorus_util,
                                 gpointer destination);
```

Set *destination* buffer of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
destination	the destination buffer	

Since: 3.13.4

ags_chorus_util_get_destination_stride ()

```
guint
ags_chorus_util_get_destination_stride
                                (AgsChorusUtil *chorus_util);
```

Get destination stride of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the destination buffer stride

Since: 3.13.4

ags_chorus_util_set_destination_stride ()

```
void
ags_chorus_util_set_destination_stride
                                (AgsChorusUtil *chorus_util,
                                 guint destination_stride);
```

Set *destination* stride of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil
destination_stride	the destination buffer stride

Since: 3.13.4

ags_chorus_util_get_buffer_length ()

```
guint
ags_chorus_util_get_buffer_length (AgsChorusUtil *chorus_util);
```

Get buffer length of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the buffer length

Since: 3.13.4

ags_chorus_util_set_buffer_length ()

```
void
ags_chorus_util_set_buffer_length (AgsChorusUtil *chorus_util,
                                   guint buffer_length);
```

Set *buffer_length* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
buffer_length	the buffer length	

Since: 3.13.4

ags_chorus_util_get_format ()

```
guint
ags_chorus_util_get_format (AgsChorusUtil *chorus_util);
```

Get format of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the format

Since: 3.13.4

ags_chorus_util_set_format ()

```
void
ags_chorus_util_set_format (AgsChorusUtil *chorus_util,
                            guint format);
```

Set *format* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
format	the format	

Since: 3.13.4

ags_chorus_util_get_samplerate ()

```
guint
ags_chorus_util_get_samplerate (AgsChorusUtil *chorus_util);
```

Get samplerate of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the samplerate

Since: 3.13.4

ags_chorus_util_set_samplerate ()

```
void
ags_chorus_util_set_samplerate (AgsChorusUtil *chorus_util,
                                guint samplerate);
```

Set *samplerate* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
samplerate	the samplerate	

Since: 3.13.4

ags_chorus_util_get_offset ()

```
guint64
ags_chorus_util_get_offset (AgsChorusUtil *chorus_util);
```

Get offset of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the offset

Since: 4.0.0

ags_chorus_util_set_offset ()

```
void
ags_chorus_util_set_offset (AgsChorusUtil *chorus_util,
                            guint64 offset);
```

Set *offset* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
offset	the offset	

Since: 4.0.0

ags_chorus_util_get_base_key ()

```
gdouble
ags_chorus_util_get_base_key (AgsChorusUtil *chorus_util);
```

Get base key of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the base key

Since: 3.13.4

ags_chorus_util_set_base_key ()

```
void
ags_chorus_util_set_base_key (AgsChorusUtil *chorus_util,
                              gdouble base_key);
```

Set *base_key* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
base_key	the base key	

Since: 3.13.4

ags_chorus_util_get_input_volume ()

```
gdouble
ags_chorus_util_get_input_volume (AgsChorusUtil *chorus_util);
```

Get input volume of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the input volume

Since: 3.13.4

ags_chorus_util_set_input_volume ()

```
void
ags_chorus_util_set_input_volume (AgsChorusUtil *chorus_util,
                                  gdouble input_volume);
```

Set *input_volume* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
input_volume	the input volume	

Since: 3.13.4

ags_chorus_util_get_output_volume ()

```
gdouble
ags_chorus_util_get_output_volume (AgsChorusUtil *chorus_util);
```

Get output volume of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the output volume

Since: 3.13.4

ags_chorus_util_set_output_volume ()

```
void
ags_chorus_util_set_output_volume (AgsChorusUtil *chorus_util,
                                   gdouble output_volume);
```

Set *output_volume* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
output_volume	the output volume	

Since: 3.13.4

ags_chorus_util_get_lfo_oscillator ()

```
guint
ags_chorus_util_get_lfo_oscillator (AgsChorusUtil *chorus_util);
```

Get LFO oscillator of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the LFO oscillator of AgsSynthOscillatorMode

Since: 3.13.4

ags_chorus_util_set_lfo_oscillator ()

```
void
ags_chorus_util_set_lfo_oscillator (AgsChorusUtil *chorus_util,
                                    guint lfo_oscillator);
```

Set *lfo_oscillator* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil
lfo_oscillator	the LFO oscillator of AgsSynthOscillatorMode

Since: 3.13.4

ags_chorus_util_get_lfo_frequency ()

```
gdouble
ags_chorus_util_get_lfo_frequency (AgsChorusUtil *chorus_util);
```

Get LFO frequency of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the LFO frequency

Since: 3.13.4

ags_chorus_util_set_lfo_frequency ()

```
void
ags_chorus_util_set_lfo_frequency (AgsChorusUtil *chorus_util,
                                   gdouble lfo_frequency);
```

Set *lfo_frequency* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
lfo_frequency	the LFO frequency	

Since: 3.13.4

ags_chorus_util_get_depth ()

```
gdouble
ags_chorus_util_get_depth (AgsChorusUtil *chorus_util);
```

Get depth of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the depth

Since: 3.13.4

ags_chorus_util_set_depth ()

```
void
ags_chorus_util_set_depth (AgsChorusUtil *chorus_util,
                           gdouble depth);
```

Set *depth* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
depth	the depth	

Since: 3.13.4

ags_chorus_util_get_mix ()

```
gdouble
ags_chorus_util_get_mix (AgsChorusUtil *chorus_util);
```

Get mix of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the mix

Since: 3.13.4

ags_chorus_util_set_mix ()

```
void
ags_chorus_util_set_mix (AgsChorusUtil *chorus_util,
                         gdouble mix);
```

Set *mix* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil
mix	the mix

Since: 3.13.4

ags_chorus_util_get_delay ()

```
gdouble
ags_chorus_util_get_delay (AgsChorusUtil *chorus_util);
```

Get delay of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Returns

the delay

Since: 3.13.4

ags_chorus_util_set_delay ()

```
void
ags_chorus_util_set_delay (AgsChorusUtil *chorus_util,
                           gdouble delay);
```

Set *delay* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
delay	the delay	

Since: 3.13.4

ags_chorus_util_get_pitch_type ()

```
GType
ags_chorus_util_get_pitch_type (AgsChorusUtil *chorus_util);
```

Get pitch type of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the pitch type

Since: 4.0.0

ags_chorus_util_set_pitch_type ()

```
void
ags_chorus_util_set_pitch_type (AgsChorusUtil *chorus_util,
                                GType pitch_type);
```

Set *pitch_type* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
pitch_type	the pitch type	

Since: 4.0.0

ags_chorus_util_get_pitch_util ()

```
gpointer
ags_chorus_util_get_pitch_util (AgsChorusUtil *chorus_util);
```

Get pitch util of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	

Returns

the pitch util.

[transfer none]

Since: 4.0.0

ags_chorus_util_set_pitch_util ()

```
void
ags_chorus_util_set_pitch_util (AgsChorusUtil *chorus_util,
                                gpointer pitch_util);
```

Set *pitch_util* of *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil	
pitch_util	the pitch util.	*[transfer full]*

Since: 4.0.0

ags_chorus_util_compute_s8 ()

```
void
ags_chorus_util_compute_s8 (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of signed 8 bit data.

Parameters

chorus_util	the AgsChorusUtil	

Since: 3.13.4

ags_chorus_util_compute_s16 ()

```
void
ags_chorus_util_compute_s16 (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of signed 16 bit data.

Parameters

chorus_util	the AgsChorusUtil	

Since: 3.13.4

ags_chorus_util_compute_s24 ()

```
void
ags_chorus_util_compute_s24 (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of signed 24 bit data.

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_compute_s32 ()

```
void
ags_chorus_util_compute_s32 (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of signed 32 bit data.

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_compute_s64 ()

```
void
ags_chorus_util_compute_s64 (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of signed 64 bit data.

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_compute_float ()

```
void
ags_chorus_util_compute_float (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of floating point data.

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_compute_double ()

```
void
ags_chorus_util_compute_double (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of double precision floating point data.

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_compute_complex ()

```
void
ags_chorus_util_compute_complex (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* of complex data.

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

ags_chorus_util_compute ()

```
void
ags_chorus_util_compute (AgsChorusUtil *chorus_util);
```

Choralize *chorus_util* .

Parameters

chorus_util	the AgsChorusUtil

Since: 3.13.4

AGS_CHORUS_UTIL()

```
#define AGS_CHORUS_UTIL(ptr) ((AgsChorusUtil *)(ptr))
```

ags_chorus_util_get_type ()

```
GType
ags_chorus_util_get_type (void);
```

Types and Values

AGS_CHORUS_UTIL_DEFAULT_PITCH_MIX_BUFFER_SIZE

```
#define AGS_CHORUS_UTIL_DEFAULT_PITCH_MIX_BUFFER_SIZE (65536)
```

AGS_CHORUS_UTIL_DEFAULT_HISTORY_BUFFER_LENGTH

```
#define AGS_CHORUS_UTIL_DEFAULT_HISTORY_BUFFER_LENGTH (256)
```

AGS_CHORUS_UTIL_DEFAULT_LFO_OSCILLATOR

```
#define AGS_CHORUS_UTIL_DEFAULT_LFO_OSCILLATOR (AGS_SYNTH_OSCILLATOR_SIN)
```

AGS_CHORUS_UTIL_DEFAULT_LFO_FREQUENCY

```
#define AGS_CHORUS_UTIL_DEFAULT_LFO_FREQUENCY (10.0)
```

AGS_CHORUS_UTIL_DEFAULT_DEPTH

```
#define AGS_CHORUS_UTIL_DEFAULT_DEPTH (0.0)
```

AGS_CHORUS_UTIL_DEFAULT_MIX

```
#define AGS_CHORUS_UTIL_DEFAULT_MIX (1.0)
```

AGS_CHORUS_UTIL_DEFAULT_DELAY

```
#define AGS_CHORUS_UTIL_DEFAULT_DELAY (0.0)
```

AGS_TYPE_CHORUS_UTIL

```
#define AGS_TYPE_CHORUS_UTIL         (ags_chorus_util_get_type())
```

struct AgsChorusUtil

```
struct AgsChorusUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  gpointer pitch_mix_buffer;
  guint pitch_mix_max_buffer_length;

  gpointer pitch_mix_buffer_history;
  guint history_buffer_length;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint64 offset;

  gdouble base_key;

  gdouble input_volume;
  gdouble output_volume;

  guint lfo_oscillator;
  gdouble lfo_frequency;

  gdouble depth;
  gdouble mix;
  gdouble delay;

  GType pitch_type;

  gpointer pitch_util;
};
```

5.7 AgsLinearInterpolateUtil

AgsLinearInterpolateUtil — util functions to linear interpolate

Stability Level

Stable, unless otherwise indicated

Functions

AgsLinearInterpolateUtil *	ags_linear_interpolate_util_alloc ()
gpointer	ags_linear_interpolate_util_copy ()
void	ags_linear_interpolate_util_free ()
gpointer	ags_linear_interpolate_util_get_source ()
void	ags_linear_interpolate_util_set_source ()
guint	ags_linear_interpolate_util_get_source_stride ()
void	ags_linear_interpolate_util_set_source_stride ()

gpointer	ags_linear_interpolate_util_get_destination ()
void	ags_linear_interpolate_util_set_destination ()
guint	ags_linear_interpolate_util_get_destination_stride ()
void	ags_linear_interpolate_util_set_destination_stride ()
guint	ags_linear_interpolate_util_get_buffer_length ()
void	ags_linear_interpolate_util_set_buffer_length ()
guint	ags_linear_interpolate_util_get_format ()
void	ags_linear_interpolate_util_set_format ()
guint	ags_linear_interpolate_util_get_samplerate ()
void	ags_linear_interpolate_util_set_samplerate ()
gdouble	ags_linear_interpolate_util_get_factor ()
void	ags_linear_interpolate_util_set_factor ()
void	ags_linear_interpolate_util_pitch_s8 ()
void	ags_linear_interpolate_util_pitch_s16 ()
void	ags_linear_interpolate_util_pitch_s24 ()
void	ags_linear_interpolate_util_pitch_s32 ()
void	ags_linear_interpolate_util_pitch_s64 ()
void	ags_linear_interpolate_util_pitch_float ()
void	ags_linear_interpolate_util_pitch_double ()
void	ags_linear_interpolate_util_pitch_complex ()
void	ags_linear_interpolate_util_pitch ()
#define	AGS_LINEAR_INTERPOLATE_UTIL()
GType	ags_linear_interpolate_util_get_type ()

Types and Values

#define	AGS_TYPE_LINEAR_INTERPOLATE_UTIL
struct	AgsLinearInterpolateUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsLinearInterpolateUtil
```

Includes

```
#include <ags/audio/ags_linear_interpolate_util.h>
```

Description

These utility functions allow you to fill linear interpolated data.

Functions

ags_linear_interpolate_util_alloc ()

```
AgsLinearInterpolateUtil~*
ags_linear_interpolate_util_alloc ();
```

Allocate AgsLinearInterpolateUtil.

Returns

the newly allocated AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_copy ()

```
gpointer
ags_linear_interpolate_util_copy (AgsLinearInterpolateUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsLinearInterpolateUtil

Returns

a pointer of the new AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_free ()

```
void
ags_linear_interpolate_util_free (AgsLinearInterpolateUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_get_source ()

```
gpointer
ags_linear_interpolate_util_get_source
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get source buffer of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Returns

the source buffer

Since: 3.9.7

ags_linear_interpolate_util_set_source ()

```
void
ags_linear_interpolate_util_set_source
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 gpointer source);
```

Set *source* buffer of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
source	the source buffer	

Since: 3.9.7

ags_linear_interpolate_util_get_source_stride ()

```
guint
ags_linear_interpolate_util_get_source_stride
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get source stride of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the source buffer stride

Since: 3.9.7

ags_linear_interpolate_util_set_source_stride ()

```
void
ags_linear_interpolate_util_set_source_stride
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 guint source_stride);
```

Set *source* stride of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
source_stride	the source buffer stride	

Since: 3.9.7

ags_linear_interpolate_util_get_destination ()

```
gpointer
ags_linear_interpolate_util_get_destination
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get destination buffer of *linear_interpolate_util*.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the destination buffer

Since: 3.9.7

ags_linear_interpolate_util_set_destination ()

```
void
ags_linear_interpolate_util_set_destination
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 gpointer destination);
```

Set *destination* buffer of *linear_interpolate_util*.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
destination	the destination buffer	

Since: 3.9.7

ags_linear_interpolate_util_get_destination_stride ()

```
guint
ags_linear_interpolate_util_get_destination_stride
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get destination stride of *linear_interpolate_util*.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the destination buffer stride

Since: 3.9.7

ags_linear_interpolate_util_set_destination_stride ()

```
void
ags_linear_interpolate_util_set_destination_stride
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 guint destination_stride);
```

Set *destination* stride of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
destination_stride	the destination buffer stride	

Since: 3.9.7

ags_linear_interpolate_util_get_buffer_length ()

```
guint
ags_linear_interpolate_util_get_buffer_length
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get buffer length of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the buffer length

Since: 3.9.7

ags_linear_interpolate_util_set_buffer_length ()

```
void
ags_linear_interpolate_util_set_buffer_length
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 guint buffer_length);
```

Set *buffer_length* of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
buffer_length	the buffer length	

Since: 3.9.7

ags_linear_interpolate_util_get_format ()

```
guint
ags_linear_interpolate_util_get_format
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get format of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the format

Since: 3.9.7

ags_linear_interpolate_util_set_format ()

```
void
ags_linear_interpolate_util_set_format
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 guint format);
```

Set *format* of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
format	the format	

Since: 3.9.7

ags_linear_interpolate_util_get_samplerate ()

```
guint
ags_linear_interpolate_util_get_samplerate
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get samplerate of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the samplerate

Since: 3.9.7

ags_linear_interpolate_util_set_samplerate ()

```
void
ags_linear_interpolate_util_set_samplerate
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 guint samplerate);
```

Set *samplerate* of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
samplerate	the samplerate	

Since: 3.9.7

ags_linear_interpolate_util_get_factor ()

```
gdouble
ags_linear_interpolate_util_get_factor
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Get factor of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Returns

the factor

Since: 3.9.7

ags_linear_interpolate_util_set_factor ()

```
void
ags_linear_interpolate_util_set_factor
                                (AgsLinearInterpolateUtil *linear_interpolate_util,
                                 gdouble factor);
```

Set *factor* of *linear_interpolate_util* .

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	
factor	the factor	

Since: 3.9.7

ags_linear_interpolate_util_pitch_s8 ()

```
void
ags_linear_interpolate_util_pitch_s8 (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of signed 8 bit data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Since: 3.9.7

ags_linear_interpolate_util_pitch_s16 ()

```
void
ags_linear_interpolate_util_pitch_s16 (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of signed 16 bit data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Since: 3.9.7

ags_linear_interpolate_util_pitch_s24 ()

```
void
ags_linear_interpolate_util_pitch_s24 (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of signed 24 bit data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil	

Since: 3.9.7

ags_linear_interpolate_util_pitch_s32 ()

```
void
ags_linear_interpolate_util_pitch_s32 (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of signed 32 bit data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_pitch_s64 ()

```
void
ags_linear_interpolate_util_pitch_s64 (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of signed 64 bit data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_pitch_float ()

```
void
ags_linear_interpolate_util_pitch_float
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of floating point data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_pitch_double ()

```
void
ags_linear_interpolate_util_pitch_double
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of double precision floating point data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_pitch_complex ()

```
void
ags_linear_interpolate_util_pitch_complex
                                (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util* of complex data.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Since: 3.9.7

ags_linear_interpolate_util_pitch ()

```
void
ags_linear_interpolate_util_pitch (AgsLinearInterpolateUtil *linear_interpolate_util);
```

Pitch *linear_interpolate_util*.

Parameters

linear_interpolate_util	the AgsLinearInterpolateUtil

Since: 3.9.7

AGS_LINEAR_INTERPOLATE_UTIL()

```
#define AGS_LINEAR_INTERPOLATE_UTIL(ptr) ((AgsLinearInterpolateUtil *)(ptr))
```

ags_linear_interpolate_util_get_type ()

```
GType
ags_linear_interpolate_util_get_type (void);
```

Types and Values

AGS_TYPE_LINEAR_INTERPOLATE_UTIL

```
#define AGS_TYPE_LINEAR_INTERPOLATE_UTIL          (ags_linear_interpolate_util_get_type())
```

struct AgsLinearInterpolateUtil

```
struct AgsLinearInterpolateUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble factor;
};
```

5.8 AgsFourierTransformUtil

AgsFourierTransformUtil — fourier transform util

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S8_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S16_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S24_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S32_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S64_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_FLOAT_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_DOUBLE_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S8_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S16_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S24_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S32_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S64_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_FLOAT_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_DOUBLE_FRAME()
#define	AGS_FOURIER_TRANSFORM_UTIL_ANALYSIS_WINDOW()
void	ags_fourier_transform_util_compute_stft_s8 ()
void	ags_fourier_transform_util_compute_stft_s16 ()
void	ags_fourier_transform_util_compute_stft_s24 ()
void	ags_fourier_transform_util_compute_stft_s32 ()
void	ags_fourier_transform_util_compute_stft_s64 ()
void	ags_fourier_transform_util_compute_stft_float ()
void	ags_fourier_transform_util_compute_stft_double ()
void	ags_fourier_transform_util_compute_stft_complex ()
void	ags_fourier_transform_util_inverse_stft_s8 ()
void	ags_fourier_transform_util_inverse_stft_s16 ()
void	ags_fourier_transform_util_inverse_stft_s24 ()
void	ags_fourier_transform_util_inverse_stft_s32 ()

void	ags_fourier_transform_util_inverse_stft_s64 ()
void	ags_fourier_transform_util_inverse_stft_float ()
void	ags_fourier_transform_util_inverse_stft_double ()
void	ags_fourier_transform_util_inverse_stft_complex ()
GType	ags_fourier_transform_util_get_type ()

Types and Values

#define	AGS_TYPE_FOURIER_TRANSFORM_UTIL
struct	AgsFourierTransformUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFourierTransformUtil
```

Includes

```
#include <ags/audio/ags_fourier_transform_util.h>
```

Description

Utility functions to compute fourier transform.

Functions

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S8_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S8_FRAME(x_buffer, x_channels, ↩
    x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S16_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S16_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S24_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S24_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S32_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S32_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S64_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S64_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_FLOAT_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_FLOAT_FRAME(x_buffer,  ↩
    x_channels, x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_DOUBLE_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_DOUBLE_FRAME(x_buffer,  ↩
    x_channels, x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S8_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S8_FRAME(x_buffer, x_channels,  ↩
    x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S16_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S16_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S24_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S24_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S32_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S32_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S64_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S64_FRAME(x_buffer, x_channels, ↩
     x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_FLOAT_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_FLOAT_FRAME(x_buffer,  ↩
    x_channels, x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_DOUBLE_FRAME()

```
#define             AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_DOUBLE_FRAME(x_buffer,  ↩
    x_channels, x_n, x_buffer_length, x_retval)
```

AGS_FOURIER_TRANSFORM_UTIL_ANALYSIS_WINDOW()

```
#define AGS_FOURIER_TRANSFORM_UTIL_ANALYSIS_WINDOW(x) (x + 1.0)
```

ags_fourier_transform_util_compute_stft_s8 ()

```
void
ags_fourier_transform_util_compute_stft_s8
                                (gint8 *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_s16 ()

```
void
ags_fourier_transform_util_compute_stft_s16
                                (gint16 *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_s24 ()

```
void
ags_fourier_transform_util_compute_stft_s24
                                (gint32 *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_s32 ()

```
void
ags_fourier_transform_util_compute_stft_s32
                                (gint32 *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_s64 ()

```
void
ags_fourier_transform_util_compute_stft_s64
                                (gint64 *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_float ()

```
void
ags_fourier_transform_util_compute_stft_float
                                (gfloat *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_double ()

```
void
ags_fourier_transform_util_compute_stft_double
                                (gdouble *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_compute_stft_complex ()

```
void
ags_fourier_transform_util_compute_stft_complex
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute fourier transform of *buffer* .

Parameters

buffer	the audio buffer	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.6.0

ags_fourier_transform_util_inverse_stft_s8 ()

```
void
ags_fourier_transform_util_inverse_stft_s8
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gint8 **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_s16 ()

```
void
ags_fourier_transform_util_inverse_stft_s16
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gint16 **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_s24 ()

```
void
ags_fourier_transform_util_inverse_stft_s24
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gint32 **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_s32 ()

```
void
ags_fourier_transform_util_inverse_stft_s32
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gint32 **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_s64 ()

```
void
ags_fourier_transform_util_inverse_stft_s64
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gint64 **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_float ()

```
void
ags_fourier_transform_util_inverse_stft_float
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gfloat **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_double ()

```
void
ags_fourier_transform_util_inverse_stft_double
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 gdouble **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.0.0

ags_fourier_transform_util_inverse_stft_complex ()

```
void
ags_fourier_transform_util_inverse_stft_complex
                                (AgsComplex *buffer,
                                 guint channels,
                                 guint buffer_length,
                                 AgsComplex **retval);
```

Compute inverse fourier transform of *buffer* .

Parameters

buffer	the complex data	
channels	number of audio channels	
buffer_length	the buffer's length	
retval	the return location of result	

Since: 3.6.0

ags_fourier_transform_util_get_type ()

```
GType
ags_fourier_transform_util_get_type (void);
```

Types and Values

AGS_TYPE_FOURIER_TRANSFORM_UTIL

```
#define AGS_TYPE_FOURIER_TRANSFORM_UTIL         (ags_fourier_transform_util_get_type())
```

struct AgsFourierTransformUtil

```
struct AgsFourierTransformUtil {
  gpointer source;
  guint source_stride;

  AgsComplex *destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;
};
```

5.9 AgsFastPitchUtil

AgsFastPitchUtil — fast pitch util

Stability Level

Stable, unless otherwise indicated

Functions

AgsFastPitchUtil *	ags_fast_pitch_util_alloc ()
gpointer	ags_fast_pitch_util_copy ()
void	ags_fast_pitch_util_free ()
gpointer	ags_fast_pitch_util_get_source ()
void	ags_fast_pitch_util_set_source ()
guint	ags_fast_pitch_util_get_source_stride ()
void	ags_fast_pitch_util_set_source_stride ()
gpointer	ags_fast_pitch_util_get_destination ()
void	ags_fast_pitch_util_set_destination ()
guint	ags_fast_pitch_util_get_destination_stride ()
void	ags_fast_pitch_util_set_destination_stride ()
guint	ags_fast_pitch_util_get_buffer_length ()
void	ags_fast_pitch_util_set_buffer_length ()
guint	ags_fast_pitch_util_get_format ()
void	ags_fast_pitch_util_set_format ()
guint	ags_fast_pitch_util_get_samplerate ()
void	ags_fast_pitch_util_set_samplerate ()
gdouble	ags_fast_pitch_util_get_base_key ()
void	ags_fast_pitch_util_set_base_key ()
gdouble	ags_fast_pitch_util_get_tuning ()
void	ags_fast_pitch_util_set_tuning ()
void	ags_fast_pitch_util_pitch_s8 ()
void	ags_fast_pitch_util_pitch_s16 ()
void	ags_fast_pitch_util_pitch_s24 ()
void	ags_fast_pitch_util_pitch_s32 ()
void	ags_fast_pitch_util_pitch_s64 ()
void	ags_fast_pitch_util_pitch_float ()
void	ags_fast_pitch_util_pitch_double ()
void	ags_fast_pitch_util_pitch_complex ()
void	ags_fast_pitch_util_pitch ()
#define	AGS_FAST_PITCH_UTIL()
GType	ags_fast_pitch_util_get_type ()

Types and Values

#define	AGS_FAST_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE
#define	AGS_TYPE_FAST_PITCH_UTIL
struct	AgsFastPitchUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsFastPitchUtil
```

Includes

```
#include <ags/audio/ags_fast_pitch_util.h>
```

Description

Utility functions to pitch.

Functions

ags_fast_pitch_util_alloc ()

```
AgsFastPitchUtil~*
ags_fast_pitch_util_alloc ();
```

Allocate AgsFastPitchUtil.

Returns

the newly allocated AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_copy ()

```
gpointer
ags_fast_pitch_util_copy (AgsFastPitchUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFastPitchUtil

Returns

a pointer of the new AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_free ()

```
void
ags_fast_pitch_util_free (AgsFastPitchUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFastPitchUtil	

Since: 3.9.6

ags_fast_pitch_util_get_source ()

```
gpointer
ags_fast_pitch_util_get_source (AgsFastPitchUtil *fast_pitch_util);
```

Get source buffer of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	

Returns

the source buffer

Since: 3.9.6

ags_fast_pitch_util_set_source ()

```
void
ags_fast_pitch_util_set_source (AgsFastPitchUtil *fast_pitch_util,
                                gpointer source);
```

Set *source* buffer of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
source	the source buffer	

Since: 3.9.6

ags_fast_pitch_util_get_source_stride ()

```
guint
ags_fast_pitch_util_get_source_stride (AgsFastPitchUtil *fast_pitch_util);
```

Get source stride of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	

Returns

the source buffer stride

Since: 3.9.6

ags_fast_pitch_util_set_source_stride ()

```
void
ags_fast_pitch_util_set_source_stride (AgsFastPitchUtil *fast_pitch_util,
                                       guint source_stride);
```

Set *source* stride of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_fast_pitch_util_get_destination ()

```
gpointer
ags_fast_pitch_util_get_destination (AgsFastPitchUtil *fast_pitch_util);
```

Get destination buffer of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	

Returns

the destination buffer

Since: 3.9.6

ags_fast_pitch_util_set_destination ()

```
void
ags_fast_pitch_util_set_destination (AgsFastPitchUtil *fast_pitch_util,
                                     gpointer destination);
```

Set *destination* buffer of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
destination	the destination buffer	

Since: 3.9.6

ags_fast_pitch_util_get_destination_stride ()

```
guint
ags_fast_pitch_util_get_destination_stride
                                (AgsFastPitchUtil *fast_pitch_util);
```

Get destination stride of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil

Returns

the destination buffer stride

Since: 3.9.6

ags_fast_pitch_util_set_destination_stride ()

```
void
ags_fast_pitch_util_set_destination_stride
                                (AgsFastPitchUtil *fast_pitch_util,
                                 guint destination_stride);
```

Set *destination* stride of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil
destination_stride	the destination buffer stride

Since: 3.9.6

ags_fast_pitch_util_get_buffer_length ()

```
guint
ags_fast_pitch_util_get_buffer_length (AgsFastPitchUtil *fast_pitch_util);
```

Get buffer length of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil

Returns

the buffer length

Since: 3.9.6

ags_fast_pitch_util_set_buffer_length ()

```
void
ags_fast_pitch_util_set_buffer_length (AgsFastPitchUtil *fast_pitch_util,
                                       guint buffer_length);
```

Set *buffer_length* of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_fast_pitch_util_get_format ()

```
guint
ags_fast_pitch_util_get_format (AgsFastPitchUtil *fast_pitch_util);
```

Get format of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	

Returns

the format

Since: 3.9.6

ags_fast_pitch_util_set_format ()

```
void
ags_fast_pitch_util_set_format (AgsFastPitchUtil *fast_pitch_util,
                                guint format);
```

Set *format* of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
format	the format	

Since: 3.9.6

ags_fast_pitch_util_get_samplerate ()

```
guint
ags_fast_pitch_util_get_samplerate (AgsFastPitchUtil *fast_pitch_util);
```

Get samplerate of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	

Returns

the samplerate

Since: 3.9.6

ags_fast_pitch_util_set_samplerate ()

```
void
ags_fast_pitch_util_set_samplerate (AgsFastPitchUtil *fast_pitch_util,
                                    guint samplerate);
```

Set *samplerate* of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_fast_pitch_util_get_base_key ()

```
gdouble
ags_fast_pitch_util_get_base_key (AgsFastPitchUtil *fast_pitch_util);
```

Get base key of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	

Returns

the base key

Since: 3.9.6

ags_fast_pitch_util_set_base_key ()

```
void
ags_fast_pitch_util_set_base_key (AgsFastPitchUtil *fast_pitch_util,
                                  gdouble base_key);
```

Set *base_key* of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil	
base_key	the base key	

Since: 3.9.6

ags_fast_pitch_util_get_tuning ()

```
gdouble
ags_fast_pitch_util_get_tuning (AgsFastPitchUtil *fast_pitch_util);
```

Get tuning of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil

Returns

the tuning

Since: 3.9.6

ags_fast_pitch_util_set_tuning ()

```
void
ags_fast_pitch_util_set_tuning (AgsFastPitchUtil *fast_pitch_util,
                                gdouble tuning);
```

Set *tuning* of *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil
tuning	the tuning

Since: 3.9.6

ags_fast_pitch_util_pitch_s8 ()

```
void
ags_fast_pitch_util_pitch_s8 (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of signed 8 bit data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_s16 ()

```
void
ags_fast_pitch_util_pitch_s16 (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of signed 16 bit data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_s24 ()

```
void
ags_fast_pitch_util_pitch_s24 (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of signed 24 bit data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_s32 ()

```
void
ags_fast_pitch_util_pitch_s32 (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of signed 32 bit data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_s64 ()

```
void
ags_fast_pitch_util_pitch_s64 (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of signed 64 bit data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_float ()

```
void
ags_fast_pitch_util_pitch_float (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of floating point data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_double ()

```
void
ags_fast_pitch_util_pitch_double (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of double precision floating point data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch_complex ()

```
void
ags_fast_pitch_util_pitch_complex (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* of complex data.

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

ags_fast_pitch_util_pitch ()

```
void
ags_fast_pitch_util_pitch (AgsFastPitchUtil *fast_pitch_util);
```

Pitch *fast_pitch_util* .

Parameters

fast_pitch_util	the AgsFastPitchUtil

Since: 3.9.6

AGS_FAST_PITCH_UTIL()

```
#define AGS_FAST_PITCH_UTIL(ptr) ((AgsFastPitchUtil *)(ptr))
```

ags_fast_pitch_util_get_type ()

```
GType
ags_fast_pitch_util_get_type (void);
```

Types and Values

AGS_FAST_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE

```
#define AGS_FAST_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE (65536)
```

AGS_TYPE_FAST_PITCH_UTIL

```
#define AGS_TYPE_FAST_PITCH_UTIL         (ags_fast_pitch_util_get_type())
```

struct AgsFastPitchUtil

```
struct AgsFastPitchUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  gpointer mix_buffer;
  guint mix_buffer_max_buffer_length;

  gpointer im_mix_buffer;
  guint im_mix_buffer_max_buffer_length;

  gpointer low_mix_buffer;
  guint low_mix_buffer_max_buffer_length;

  gpointer new_mix_buffer;
  guint new_mix_buffer_max_buffer_length;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble base_key;
  gdouble tuning;
};
```

5.10 AgsHQPitchUtil

AgsHQPitchUtil — hq pitch util

Stability Level

Stable, unless otherwise indicated

Functions

AgsHQPitchUtil *	ags_hq_pitch_util_alloc ()
gpointer	ags_hq_pitch_util_copy ()
void	ags_hq_pitch_util_free ()
gpointer	ags_hq_pitch_util_get_source ()
void	ags_hq_pitch_util_set_source ()
guint	ags_hq_pitch_util_get_source_stride ()
void	ags_hq_pitch_util_set_source_stride ()
gpointer	ags_hq_pitch_util_get_destination ()
void	ags_hq_pitch_util_set_destination ()
guint	ags_hq_pitch_util_get_destination_stride ()
void	ags_hq_pitch_util_set_destination_stride ()
guint	ags_hq_pitch_util_get_buffer_length ()
void	ags_hq_pitch_util_set_buffer_length ()
guint	ags_hq_pitch_util_get_format ()
void	ags_hq_pitch_util_set_format ()
guint	ags_hq_pitch_util_get_samplerate ()
void	ags_hq_pitch_util_set_samplerate ()
gdouble	ags_hq_pitch_util_get_base_key ()
void	ags_hq_pitch_util_set_base_key ()
gdouble	ags_hq_pitch_util_get_tuning ()
void	ags_hq_pitch_util_set_tuning ()
void	ags_hq_pitch_util_pitch_s8 ()
void	ags_hq_pitch_util_pitch_s16 ()
void	ags_hq_pitch_util_pitch_s24 ()
void	ags_hq_pitch_util_pitch_s32 ()
void	ags_hq_pitch_util_pitch_s64 ()
void	ags_hq_pitch_util_pitch_float ()
void	ags_hq_pitch_util_pitch_double ()
void	ags_hq_pitch_util_pitch_complex ()
void	ags_hq_pitch_util_pitch ()
#define	AGS_HQ_PITCH_UTIL()
GType	ags_hq_pitch_util_get_type ()

Types and Values

#define	AGS_HQ_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE
#define	AGS_TYPE_HQ_PITCH_UTIL
struct	AgsHQPitchUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsHQPitchUtil
```

Includes

```
#include <ags/audio/ags_hq_pitch_util.h>
```

Description

Utility functions to pitch.

Functions

ags_hq_pitch_util_alloc ()

```
AgsHQPitchUtil~*
ags_hq_pitch_util_alloc ();
```

Allocate AgsHQPitchUtil.

Returns

the newly allocated AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_copy ()

```
gpointer
ags_hq_pitch_util_copy (AgsHQPitchUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsHQPitchUtil

Returns

a pointer of the new AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_free ()

```
void
ags_hq_pitch_util_free (AgsHQPitchUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_get_source ()

```
gpointer
ags_hq_pitch_util_get_source (AgsHQPitchUtil *hq_pitch_util);
```

Get source buffer of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the source buffer

Since: 3.9.6

ags_hq_pitch_util_set_source ()

```
void
ags_hq_pitch_util_set_source (AgsHQPitchUtil *hq_pitch_util,
                              gpointer source);
```

Set *source* buffer of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
source	the source buffer	

Since: 3.9.6

ags_hq_pitch_util_get_source_stride ()

```
guint
ags_hq_pitch_util_get_source_stride (AgsHQPitchUtil *hq_pitch_util);
```

Get source stride of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the source buffer stride

Since: 3.9.6

ags_hq_pitch_util_set_source_stride ()

```
void
ags_hq_pitch_util_set_source_stride (AgsHQPitchUtil *hq_pitch_util,
                                     guint source_stride);
```

Set *source* stride of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_hq_pitch_util_get_destination ()

```
gpointer
ags_hq_pitch_util_get_destination (AgsHQPitchUtil *hq_pitch_util);
```

Get destination buffer of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the destination buffer

Since: 3.9.6

ags_hq_pitch_util_set_destination ()

```
void
ags_hq_pitch_util_set_destination (AgsHQPitchUtil *hq_pitch_util,
                                   gpointer destination);
```

Set *destination* buffer of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
destination	the destination buffer	

Since: 3.9.6

ags_hq_pitch_util_get_destination_stride ()

```
guint
ags_hq_pitch_util_get_destination_stride
                                (AgsHQPitchUtil *hq_pitch_util);
```

Get destination stride of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the destination buffer stride

Since: 3.9.6

ags_hq_pitch_util_set_destination_stride ()

```
void
ags_hq_pitch_util_set_destination_stride
                                (AgsHQPitchUtil *hq_pitch_util,
                                 guint destination_stride);
```

Set *destination* stride of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
destination_stride	the destination buffer stride	

Since: 3.9.6

ags_hq_pitch_util_get_buffer_length ()

```
guint
ags_hq_pitch_util_get_buffer_length (AgsHQPitchUtil *hq_pitch_util);
```

Get buffer length of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the buffer length

Since: 3.9.6

ags_hq_pitch_util_set_buffer_length ()

```
void
ags_hq_pitch_util_set_buffer_length (AgsHQPitchUtil *hq_pitch_util,
                                     guint buffer_length);
```

Set *buffer_length* of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_hq_pitch_util_get_format ()

```
guint
ags_hq_pitch_util_get_format (AgsHQPitchUtil *hq_pitch_util);
```

Get format of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the format

Since: 3.9.6

ags_hq_pitch_util_set_format ()

```
void
ags_hq_pitch_util_set_format (AgsHQPitchUtil *hq_pitch_util,
                              guint format);
```

Set *format* of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
format	the format	

Since: 3.9.6

ags_hq_pitch_util_get_samplerate ()

```
guint
ags_hq_pitch_util_get_samplerate (AgsHQPitchUtil *hq_pitch_util);
```

Get samplerate of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the samplerate

Since: 3.9.6

ags_hq_pitch_util_set_samplerate ()

```
void
ags_hq_pitch_util_set_samplerate (AgsHQPitchUtil *hq_pitch_util,
                                  guint samplerate);
```

Set *samplerate* of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_hq_pitch_util_get_base_key ()

```
gdouble
ags_hq_pitch_util_get_base_key (AgsHQPitchUtil *hq_pitch_util);
```

Get base key of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the base key

Since: 3.9.6

ags_hq_pitch_util_set_base_key ()

```
void
ags_hq_pitch_util_set_base_key (AgsHQPitchUtil *hq_pitch_util,
                                gdouble base_key);
```

Set *base_key* of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
base_key	the base key	

Since: 3.9.6

ags_hq_pitch_util_get_tuning ()

```
gdouble
ags_hq_pitch_util_get_tuning (AgsHQPitchUtil *hq_pitch_util);
```

Get tuning of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Returns

the tuning

Since: 3.9.6

ags_hq_pitch_util_set_tuning ()

```
void
ags_hq_pitch_util_set_tuning (AgsHQPitchUtil *hq_pitch_util,
                              gdouble tuning);
```

Set *tuning* of *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil	
tuning	the tuning	

Since: 3.9.6

ags_hq_pitch_util_pitch_s8 ()

```
void
ags_hq_pitch_util_pitch_s8 (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of signed 8 bit data.

Parameters

hq_pitch_util	the AgsHQPitchUtil	

Since: 3.9.6

ags_hq_pitch_util_pitch_s16 ()

```
void
ags_hq_pitch_util_pitch_s16 (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of signed 16 bit data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch_s24 ()

```
void
ags_hq_pitch_util_pitch_s24 (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of signed 24 bit data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch_s32 ()

```
void
ags_hq_pitch_util_pitch_s32 (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of signed 32 bit data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch_s64 ()

```
void
ags_hq_pitch_util_pitch_s64 (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of signed 64 bit data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch_float ()

```
void
ags_hq_pitch_util_pitch_float (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of floating point data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch_double ()

```
void
ags_hq_pitch_util_pitch_double (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of double precision floating point data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch_complex ()

```
void
ags_hq_pitch_util_pitch_complex (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* of complex data.

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

ags_hq_pitch_util_pitch ()

```
void
ags_hq_pitch_util_pitch (AgsHQPitchUtil *hq_pitch_util);
```

Pitch *hq_pitch_util* .

Parameters

hq_pitch_util	the AgsHQPitchUtil

Since: 3.9.6

AGS_HQ_PITCH_UTIL()

```
#define AGS_HQ_PITCH_UTIL(ptr) ((AgsHQPitchUtil *)(ptr))
```

ags_hq_pitch_util_get_type ()

```
GType
ags_hq_pitch_util_get_type (void);
```

Types and Values

AGS_HQ_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE

```
#define AGS_HQ_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE (65536)
```

AGS_TYPE_HQ_PITCH_UTIL

```
#define AGS_TYPE_HQ_PITCH_UTIL         (ags_hq_pitch_util_get_type())
```

struct AgsHQPitchUtil

```
struct AgsHQPitchUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  gpointer low_mix_buffer;
  guint low_mix_buffer_max_buffer_length;

  gpointer new_mix_buffer;
  guint new_mix_buffer_max_buffer_length;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble base_key;
  gdouble tuning;

  AgsLinearInterpolateUtil *linear_interpolate_util;
};
```

5.11 AgsTimeStretchUtil

AgsTimeStretchUtil — time stretch util

Stability Level

Stable, unless otherwise indicated

Functions

AgsTimeStretchUtil *	ags_time_stretch_util_alloc ()
gpointer	ags_time_stretch_util_copy ()
void	ags_time_stretch_util_free ()
gpointer	ags_time_stretch_util_get_source ()
void	ags_time_stretch_util_set_source ()
guint	ags_time_stretch_util_get_source_stride ()
void	ags_time_stretch_util_set_source_stride ()
guint	ags_time_stretch_util_get_source_buffer_length ()
void	ags_time_stretch_util_set_source_buffer_length ()
gpointer	ags_time_stretch_util_get_destination ()
void	ags_time_stretch_util_set_destination ()
guint	ags_time_stretch_util_get_destination_stride ()
void	ags_time_stretch_util_set_destination_stride ()
guint	ags_time_stretch_util_get_destination_buffer_length ()
void	ags_time_stretch_util_set_destination_buffer_length ()
guint	ags_time_stretch_util_get_buffer_size ()
void	ags_time_stretch_util_set_buffer_size ()
guint	ags_time_stretch_util_get_format ()
void	ags_time_stretch_util_set_format ()
guint	ags_time_stretch_util_get_samplerate ()
void	ags_time_stretch_util_set_samplerate ()
gdouble	ags_time_stretch_util_get_frequency ()
void	ags_time_stretch_util_set_frequency ()
gdouble	ags_time_stretch_util_get_orig_bpm ()
void	ags_time_stretch_util_set_orig_bpm ()
gdouble	ags_time_stretch_util_get_new_bpm ()
void	ags_time_stretch_util_set_new_bpm ()
void	ags_time_stretch_util_stretch_s8 ()
void	ags_time_stretch_util_stretch_s16 ()
void	ags_time_stretch_util_stretch_s24 ()
void	ags_time_stretch_util_stretch_s32 ()
void	ags_time_stretch_util_stretch_s64 ()
void	ags_time_stretch_util_stretch_float ()
void	ags_time_stretch_util_stretch_double ()
void	ags_time_stretch_util_stretch_complex ()
void	ags_time_stretch_util_stretch ()
#define	AGS_TIME_STRETCH_UTIL()
GType	ags_time_stretch_util_get_type ()

Types and Values

#define	AGS_TIME_STRETCH_UTIL_DEFAULT_FREQUENCY
#define	AGS_TYPE_TIME_STRETCH_UTIL
struct	AgsTimeStretchUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsTimeStretchUtil
```

Includes

```
#include <ags/audio/ags_time_stretch_util.h>
```

Description

Utility functions to stretch time.

Functions

ags_time_stretch_util_alloc ()

```
AgsTimeStretchUtil~*
ags_time_stretch_util_alloc ();
```

Allocate AgsTimeStretchUtil.

Returns

the newly allocated AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_copy ()

```
gpointer
ags_time_stretch_util_copy (AgsTimeStretchUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsTimeStretchUtil

Returns

a pointer of the new AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_free ()

```
void
ags_time_stretch_util_free (AgsTimeStretchUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsTimeStretchUtil	

Since: 3.13.0

ags_time_stretch_util_get_source ()

```
gpointer
ags_time_stretch_util_get_source (AgsTimeStretchUtil *time_stretch_util);
```

Get source buffer of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the source buffer

Since: 3.13.0

ags_time_stretch_util_set_source ()

```
void
ags_time_stretch_util_set_source (AgsTimeStretchUtil *time_stretch_util,
                                  gpointer source);
```

Set *source* buffer of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
source	the source buffer	

Since: 3.13.0

ags_time_stretch_util_get_source_stride ()

```
guint
ags_time_stretch_util_get_source_stride
                                (AgsTimeStretchUtil *time_stretch_util);
```

Get source stride of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the source buffer stride

Since: 3.13.0

ags_time_stretch_util_set_source_stride ()

```
void
ags_time_stretch_util_set_source_stride
                                (AgsTimeStretchUtil *time_stretch_util,
                                 guint source_stride);
```

Set *source* stride of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
source_stride	the source buffer stride	

Since: 3.13.0

ags_time_stretch_util_get_source_buffer_length ()

```
guint
ags_time_stretch_util_get_source_buffer_length
                                (AgsTimeStretchUtil *time_stretch_util);
```

Get source buffer length of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the source buffer length

Since: 3.13.0

ags_time_stretch_util_set_source_buffer_length ()

```
void
ags_time_stretch_util_set_source_buffer_length
                                (AgsTimeStretchUtil *time_stretch_util,
                                 guint source_buffer_length);
```

Set *source_buffer_length* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
source_buffer_length	the buffer length	

Since: 3.13.0

ags_time_stretch_util_get_destination ()

```
gpointer
ags_time_stretch_util_get_destination (AgsTimeStretchUtil *time_stretch_util);
```

Get destination buffer of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil

Returns

the destination buffer

Since: 3.13.0

ags_time_stretch_util_set_destination ()

```
void
ags_time_stretch_util_set_destination (AgsTimeStretchUtil *time_stretch_util,
                                       gpointer destination);
```

Set *destination* buffer of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil
destination	the destination buffer

Since: 3.13.0

ags_time_stretch_util_get_destination_stride ()

```
guint
ags_time_stretch_util_get_destination_stride
                                (AgsTimeStretchUtil *time_stretch_util);
```

Get destination stride of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil

Returns

the destination buffer stride

Since: 3.13.0

ags_time_stretch_util_set_destination_stride ()

```
void
ags_time_stretch_util_set_destination_stride
                                (AgsTimeStretchUtil *time_stretch_util,
                                 guint destination_stride);
```

Set *destination* stride of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
destination_stride	the destination buffer stride	

Since: 3.13.0

ags_time_stretch_util_get_destination_buffer_length ()

```
guint
ags_time_stretch_util_get_destination_buffer_length
                                (AgsTimeStretchUtil *time_stretch_util);
```

Get destination buffer length of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the destination buffer length

Since: 3.13.0

ags_time_stretch_util_set_destination_buffer_length ()

```
void
ags_time_stretch_util_set_destination_buffer_length
                                (AgsTimeStretchUtil *time_stretch_util,
                                 guint destination_buffer_length);
```

Set *destination_buffer_length* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
destination_buffer_length	the buffer length	

Since: 3.13.0

ags_time_stretch_util_get_buffer_size ()

```
guint
ags_time_stretch_util_get_buffer_size (AgsTimeStretchUtil *time_stretch_util);
```

Get buffer size of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the buffer size

Since: 3.13.0

ags_time_stretch_util_set_buffer_size ()

```
void
ags_time_stretch_util_set_buffer_size (AgsTimeStretchUtil *time_stretch_util,
                                       guint buffer_size);
```

Set *buffer_size* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
buffer_size	the buffer size	

Since: 3.13.0

ags_time_stretch_util_get_format ()

```
guint
ags_time_stretch_util_get_format (AgsTimeStretchUtil *time_stretch_util);
```

Get format of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the format

Since: 3.13.0

ags_time_stretch_util_set_format ()

```
void
ags_time_stretch_util_set_format (AgsTimeStretchUtil *time_stretch_util,
                                  guint format);
```

Set *format* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
format	the format	

Since: 3.13.0

ags_time_stretch_util_get_samplerate ()

```
guint
ags_time_stretch_util_get_samplerate (AgsTimeStretchUtil *time_stretch_util);
```

Get samplerate of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the samplerate

Since: 3.13.0

ags_time_stretch_util_set_samplerate ()

```
void
ags_time_stretch_util_set_samplerate (AgsTimeStretchUtil *time_stretch_util,
                                      guint samplerate);
```

Set *samplerate* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
samplerate	the samplerate	

Since: 3.13.0

ags_time_stretch_util_get_frequency ()

```
gdouble
ags_time_stretch_util_get_frequency (AgsTimeStretchUtil *time_stretch_util);
```

Get frequency of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the frequency

Since: 3.13.0

ags_time_stretch_util_set_frequency ()

```
void
ags_time_stretch_util_set_frequency (AgsTimeStretchUtil *time_stretch_util,
                                     gdouble frequency);
```

Set *frequency* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
frequency	the frequency	

Since: 3.13.0

ags_time_stretch_util_get_orig_bpm ()

```
gdouble
ags_time_stretch_util_get_orig_bpm (AgsTimeStretchUtil *time_stretch_util);
```

Get original BPM of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the original BPM

Since: 3.13.0

ags_time_stretch_util_set_orig_bpm ()

```
void
ags_time_stretch_util_set_orig_bpm (AgsTimeStretchUtil *time_stretch_util,
                                    gdouble orig_bpm);
```

Set *orig_bpm* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
orig_bpm	the original BPM	

Since: 3.13.0

ags_time_stretch_util_get_new_bpm ()

```
gdouble
ags_time_stretch_util_get_new_bpm (AgsTimeStretchUtil *time_stretch_util);
```

Get new BPM of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Returns

the new BPM

Since: 3.13.0

ags_time_stretch_util_set_new_bpm ()

```
void
ags_time_stretch_util_set_new_bpm (AgsTimeStretchUtil *time_stretch_util,
                                   gdouble new_bpm);
```

Set *new_bpm* of *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil	
new_bpm	the new_bpm	

Since: 3.13.0

ags_time_stretch_util_stretch_s8 ()

```
void
ags_time_stretch_util_stretch_s8 (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of signed 8 bit data.

Parameters

time_stretch_util	the AgsTimeStretchUtil	

Since: 3.13.0

ags_time_stretch_util_stretch_s16 ()

```
void
ags_time_stretch_util_stretch_s16 (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of signed 16 bit data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch_s24 ()

```
void
ags_time_stretch_util_stretch_s24 (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of signed 24 bit data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch_s32 ()

```
void
ags_time_stretch_util_stretch_s32 (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of signed 32 bit data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch_s64 ()

```
void
ags_time_stretch_util_stretch_s64 (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of signed 64 bit data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch_float ()

```
void
ags_time_stretch_util_stretch_float (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of floating point data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch_double ()

```
void
ags_time_stretch_util_stretch_double (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of double precision floating point data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch_complex ()

```
void
ags_time_stretch_util_stretch_complex (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* of complex data.

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

ags_time_stretch_util_stretch ()

```
void
ags_time_stretch_util_stretch (AgsTimeStretchUtil *time_stretch_util);
```

Stretch *time_stretch_util* .

Parameters

time_stretch_util	the AgsTimeStretchUtil

Since: 3.13.0

AGS_TIME_STRETCH_UTIL()

```
#define AGS_TIME_STRETCH_UTIL(ptr) ((AgsTimeStretchUtil *)(ptr))
```

ags_time_stretch_util_get_type ()

```
GType
ags_time_stretch_util_get_type (void);
```

Types and Values

AGS_TIME_STRETCH_UTIL_DEFAULT_FREQUENCY

```
#define AGS_TIME_STRETCH_UTIL_DEFAULT_FREQUENCY (440.0)
```

AGS_TYPE_TIME_STRETCH_UTIL

```
#define AGS_TYPE_TIME_STRETCH_UTIL          (ags_time_stretch_util_get_type())
```

struct AgsTimeStretchUtil

```
struct AgsTimeStretchUtil {
  gpointer source;
  guint source_stride;
  guint source_buffer_length;

  gpointer destination;
  guint destination_stride;
  guint destination_buffer_length;

  guint buffer_size;
  guint format;
  guint samplerate;

  gdouble frequency;

  gdouble orig_bpm;
  gdouble new_bpm;

};
```

5.12 AgsPhaseShiftUtil

AgsPhaseShiftUtil — phase shift util

Stability Level

Stable, unless otherwise indicated

Functions

AgsPhaseShiftUtil *	ags_phase_shift_util_alloc ()
gpointer	ags_phase_shift_util_copy ()
void	ags_phase_shift_util_free ()
gpointer	ags_phase_shift_util_get_destination ()
void	ags_phase_shift_util_set_destination ()
guint	ags_phase_shift_util_get_destination_stride ()
void	ags_phase_shift_util_set_destination_stride ()
gpointer	ags_phase_shift_util_get_source ()
void	ags_phase_shift_util_set_source ()
guint	ags_phase_shift_util_get_source_stride ()
void	ags_phase_shift_util_set_source_stride ()
guint	ags_phase_shift_util_get_buffer_length ()
void	ags_phase_shift_util_set_buffer_length ()
guint	ags_phase_shift_util_get_format ()
void	ags_phase_shift_util_set_format ()
gdouble	ags_phase_shift_util_get_frequency ()
void	ags_phase_shift_util_set_frequency ()
gdouble	ags_phase_shift_util_get_amount ()
void	ags_phase_shift_util_set_amount ()
gdouble	ags_phase_shift_util_get_phase ()
void	ags_phase_shift_util_set_phase ()
void	ags_phase_shift_util_process_s8 ()
void	ags_phase_shift_util_process_s16 ()
void	ags_phase_shift_util_process_s24 ()
void	ags_phase_shift_util_process_s32 ()
void	ags_phase_shift_util_process_s64 ()
void	ags_phase_shift_util_process_float ()
void	ags_phase_shift_util_process_double ()
void	ags_phase_shift_util_process_complex ()
void	ags_phase_shift_util_process ()
GType	ags_phase_shift_util_get_type ()

Types and Values

#define	AGS_PHASE_SHIFT_UTIL_DEFAULT_FORMAT
#define	AGS_TYPE_PHASE_SHIFT_UTIL
struct	AgsPhaseShiftUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsPhaseShiftUtil
```

Includes

```
#include <ags/audio/ags_phase_shift_util.h>
```

Description

Utility functions to compute phase shift.

Functions

ags_phase_shift_util_alloc ()

```
AgsPhaseShiftUtil~*
ags_phase_shift_util_alloc ();
```

Allocate AgsPhaseShiftUtil

Returns

a new AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_copy ()

```
gpointer
ags_phase_shift_util_copy (AgsPhaseShiftUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsPhaseShiftUtil

Returns

a pointer of the new AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_free ()

```
void
ags_phase_shift_util_free (AgsPhaseShiftUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_get_destination ()

```
gpointer
ags_phase_shift_util_get_destination (AgsPhaseShiftUtil *phase_shift_util);
```

Get destination buffer of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Returns

the destination buffer

Since: 4.0.0

ags_phase_shift_util_set_destination ()

```
void
ags_phase_shift_util_set_destination (AgsPhaseShiftUtil *phase_shift_util,
                                      gpointer destination);
```

Set *destination* buffer of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil
destination	the destination buffer

Since: 4.0.0

ags_phase_shift_util_get_destination_stride ()

```
guint
ags_phase_shift_util_get_destination_stride
                                (AgsPhaseShiftUtil *phase_shift_util);
```

Get destination stride of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Returns

the destination buffer stride

Since: 4.0.0

ags_phase_shift_util_set_destination_stride ()

```
void
ags_phase_shift_util_set_destination_stride
                                (AgsPhaseShiftUtil *phase_shift_util,
                                 guint destination_stride);
```

Set *destination* stride of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	
destination_stride	the destination buffer stride	

Since: 4.0.0

ags_phase_shift_util_get_source ()

```
gpointer
ags_phase_shift_util_get_source (AgsPhaseShiftUtil *phase_shift_util);
```

Get source buffer of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	

Returns

the source buffer

Since: 4.0.0

ags_phase_shift_util_set_source ()

```
void
ags_phase_shift_util_set_source (AgsPhaseShiftUtil *phase_shift_util,
                                 gpointer source);
```

Set *source* buffer of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	
source	the source buffer	

Since: 4.0.0

ags_phase_shift_util_get_source_stride ()

```
guint
ags_phase_shift_util_get_source_stride
                                (AgsPhaseShiftUtil *phase_shift_util);
```

Get source stride of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Returns

the source buffer stride

Since: 4.0.0

ags_phase_shift_util_set_source_stride ()

```
void
ags_phase_shift_util_set_source_stride
                                (AgsPhaseShiftUtil *phase_shift_util,
                                 guint source_stride);
```

Set *source* stride of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil
source_stride	the source buffer stride

Since: 4.0.0

ags_phase_shift_util_get_buffer_length ()

```
guint
ags_phase_shift_util_get_buffer_length
                                (AgsPhaseShiftUtil *phase_shift_util);
```

Get buffer length of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Returns

the buffer length

Since: 4.0.0

ags_phase_shift_util_set_buffer_length ()

```
void
ags_phase_shift_util_set_buffer_length
                                (AgsPhaseShiftUtil *phase_shift_util,
                                 guint buffer_length);
```

Set *buffer_length* of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	
buffer_length	the buffer length	

Since: 4.0.0

ags_phase_shift_util_get_format ()

```
guint
ags_phase_shift_util_get_format (AgsPhaseShiftUtil *phase_shift_util);
```

Get format of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	

Returns

the format

Since: 4.0.0

ags_phase_shift_util_set_format ()

```
void
ags_phase_shift_util_set_format (AgsPhaseShiftUtil *phase_shift_util,
                                 guint format);
```

Set *format* of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	
format	the format	

Since: 4.0.0

ags_phase_shift_util_get_frequency ()

```
gdouble
ags_phase_shift_util_get_frequency (AgsPhaseShiftUtil *phase_shift_util);
```

Get frequency of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Returns

the frequency

Since: 4.0.0

ags_phase_shift_util_set_frequency ()

```
void
ags_phase_shift_util_set_frequency (AgsPhaseShiftUtil *phase_shift_util,
                                    gdouble frequency);
```

Set *frequency* of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil
frequency	the frequency

Since: 4.0.0

ags_phase_shift_util_get_amount ()

```
gdouble
ags_phase_shift_util_get_amount (AgsPhaseShiftUtil *phase_shift_util);
```

Get amount of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Returns

the amount

Since: 4.0.0

ags_phase_shift_util_set_amount ()

```
void
ags_phase_shift_util_set_amount (AgsPhaseShiftUtil *phase_shift_util,
                                 gdouble amount);
```

Set *amount* of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	
amount	the amount	

Since: 4.0.0

ags_phase_shift_util_get_phase ()

```
gdouble
ags_phase_shift_util_get_phase (AgsPhaseShiftUtil *phase_shift_util);
```

Get phase of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	

Returns

the phase

Since: 4.0.0

ags_phase_shift_util_set_phase ()

```
void
ags_phase_shift_util_set_phase (AgsPhaseShiftUtil *phase_shift_util,
                                gdouble phase);
```

Set *phase* of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil	
phase	the phase	

Since: 4.0.0

ags_phase_shift_util_process_s8 ()

```
void
ags_phase_shift_util_process_s8 (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of signed 8 bit data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_s16 ()

```
void
ags_phase_shift_util_process_s16 (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of signed 16 bit data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_s24 ()

```
void
ags_phase_shift_util_process_s24 (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of signed 24 bit data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_s32 ()

```
void
ags_phase_shift_util_process_s32 (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of signed 32 bit data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_s64 ()

```
void
ags_phase_shift_util_process_s64 (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of signed 64 bit data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_float ()

```
void
ags_phase_shift_util_process_float (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of single precision floating point data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_double ()

```
void
ags_phase_shift_util_process_double (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* of double precision floating point data.

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process_complex ()

```
void
ags_phase_shift_util_process_complex (AgsPhaseShiftUtil *phase_shift_util);
```

Process phase shift of *phase_shift_util* .

Parameters

phase_shift_util	the AgsPhaseShiftUtil

Since: 4.0.0

ags_phase_shift_util_process ()

```
void
ags_phase_shift_util_process (AgsPhaseShiftUtil *phase_shift_util);
```

ags_phase_shift_util_get_type ()

```
GType
ags_phase_shift_util_get_type (void);
```

Types and Values

AGS_PHASE_SHIFT_UTIL_DEFAULT_FORMAT

```
#define AGS_PHASE_SHIFT_UTIL_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_TYPE_PHASE_SHIFT_UTIL

```
#define AGS_TYPE_PHASE_SHIFT_UTIL         (ags_phase_shift_util_get_type())
```

struct AgsPhaseShiftUtil

```
struct AgsPhaseShiftUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble frequency;
  gdouble amount;
  gdouble phase;
};
```

5.13 AgsFrequencyAliaseUtil

AgsFrequencyAliaseUtil — frequency aliase util

Stability Level

Stable, unless otherwise indicated

Functions

AgsFrequencyAliaseUtil *	ags_frequency_aliase_util_alloc ()
gpointer	ags_frequency_aliase_util_copy ()
void	ags_frequency_aliase_util_free ()
gpointer	ags_frequency_aliase_util_get_destination ()
void	ags_frequency_aliase_util_set_destination ()

guint	ags_frequency_aliase_util_get_destination_stride ()
void	ags_frequency_aliase_util_set_destination_stride ()
gpointer	ags_frequency_aliase_util_get_source ()
void	ags_frequency_aliase_util_set_source ()
guint	ags_frequency_aliase_util_get_source_stride ()
void	ags_frequency_aliase_util_set_source_stride ()
gpointer	ags_frequency_aliase_util_get_phase_shifted_source ()
void	ags_frequency_aliase_util_set_phase_shifted_source ()
guint	ags_frequency_aliase_util_get_phase_shifted_source_stride ()
void	ags_frequency_aliase_util_set_phase_shifted_source_stride ()
guint	ags_frequency_aliase_util_get_buffer_length ()
void	ags_frequency_aliase_util_set_buffer_length ()
guint	ags_frequency_aliase_util_get_format ()
void	ags_frequency_aliase_util_set_format ()
void	ags_frequency_aliase_util_process_s8 ()
void	ags_frequency_aliase_util_process_s16 ()
void	ags_frequency_aliase_util_process_s24 ()
void	ags_frequency_aliase_util_process_s32 ()
void	ags_frequency_aliase_util_process_s64 ()
void	ags_frequency_aliase_util_process_float ()
void	ags_frequency_aliase_util_process_double ()
void	ags_frequency_aliase_util_process_complex ()
void	ags_frequency_aliase_util_process ()
GType	ags_frequency_aliase_util_get_type ()

Types and Values

#define	AGS_FREQUENCY_ALIASE_UTIL_DEFAULT_FORMAT
#define	AGS_TYPE_FREQUENCY_ALIASE_UTIL
struct	AgsFrequencyAliaseUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFrequencyAliaseUtil
```

Includes

```
#include <ags/audio/ags_frequency_aliase_util.h>
```

Description

Utility functions to compute frequency aliase.

Functions

ags_frequency_aliase_util_alloc ()

```
AgsFrequencyAliaseUtil~*
ags_frequency_aliase_util_alloc ();
```

Allocate AgsFrequencyAliaseUtil

Returns

a new AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_copy ()

```
gpointer
ags_frequency_aliase_util_copy (AgsFrequencyAliaseUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFrequencyAliaseUtil

Returns

a pointer of the new AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_free ()

```
void
ags_frequency_aliase_util_free (AgsFrequencyAliaseUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_get_destination ()

```
gpointer
ags_frequency_aliase_util_get_destination
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get destination buffer of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Returns

the destination buffer

Since: 4.0.0

ags_frequency_aliase_util_set_destination ()

```
void
ags_frequency_aliase_util_set_destination
                                (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                 gpointer destination);
```

Set *destination* buffer of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
destination	the destination buffer	

Since: 4.0.0

ags_frequency_aliase_util_get_destination_stride ()

```
guint
ags_frequency_aliase_util_get_destination_stride
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get destination stride of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	

Returns

the destination buffer stride

Since: 4.0.0

ags_frequency_aliase_util_set_destination_stride ()

```
void
ags_frequency_aliase_util_set_destination_stride
                                (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                 guint destination_stride);
```

Set *destination* stride of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
destination_stride	the destination buffer stride	

Since: 4.0.0

ags_frequency_aliase_util_get_source ()

```
gpointer
ags_frequency_aliase_util_get_source (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get source buffer of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	

Returns

the source buffer

Since: 4.0.0

ags_frequency_aliase_util_set_source ()

```
void
ags_frequency_aliase_util_set_source (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                      gpointer source);
```

Set *source* buffer of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
source	the source buffer	

Since: 4.0.0

ags_frequency_aliase_util_get_source_stride ()

```
guint
ags_frequency_aliase_util_get_source_stride
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get source stride of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	

Returns

the source buffer stride

Since: 4.0.0

ags_frequency_aliase_util_set_source_stride ()

```
void
ags_frequency_aliase_util_set_source_stride
                                (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                 guint source_stride);
```

Set *source* stride of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
source_stride	the source buffer stride	

Since: 4.0.0

ags_frequency_aliase_util_get_phase_shifted_source ()

```
gpointer
ags_frequency_aliase_util_get_phase_shifted_source
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get phase shifted source buffer of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	

Returns

the phase shifted source buffer

Since: 4.0.0

ags_frequency_aliase_util_set_phase_shifted_source ()

```
void
ags_frequency_aliase_util_set_phase_shifted_source
                                (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                 gpointer phase_shifted_source);
```

Set *phase_shifted_source* buffer of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
phase_shifted_source	the phase shifted source buffer	

Since: 4.0.0

ags_frequency_aliase_util_get_phase_shifted_source_stride ()

```
guint
ags_frequency_aliase_util_get_phase_shifted_source_stride
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

ags_frequency_aliase_util_set_phase_shifted_source_stride ()

```
void
ags_frequency_aliase_util_set_phase_shifted_source_stride
                                (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                 guint phase_shifted_source_stride);
```

ags_frequency_aliase_util_get_buffer_length ()

```
guint
ags_frequency_aliase_util_get_buffer_length
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get buffer length of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	

Returns

the buffer length

Since: 4.0.0

ags_frequency_aliase_util_set_buffer_length ()

```
void
ags_frequency_aliase_util_set_buffer_length
                                (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                 guint buffer_length);
```

Set *buffer_length* of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
buffer_length	the buffer length	

Since: 4.0.0

ags_frequency_aliase_util_get_format ()

```
guint
ags_frequency_aliase_util_get_format (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Get format of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	

Returns

the format

Since: 4.0.0

ags_frequency_aliase_util_set_format ()

```
void
ags_frequency_aliase_util_set_format (AgsFrequencyAliaseUtil *frequency_aliase_util,
                                      guint format);
```

Set *format* of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil	
format	the format	

Since: 4.0.0

ags_frequency_aliase_util_process_s8 ()

```
void
ags_frequency_aliase_util_process_s8 (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of signed 8 bit data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_s16 ()

```
void
ags_frequency_aliase_util_process_s16 (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of signed 16 bit data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_s24 ()

```
void
ags_frequency_aliase_util_process_s24 (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of signed 24 bit data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_s32 ()

```
void
ags_frequency_aliase_util_process_s32 (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of signed 32 bit data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_s64 ()

```
void
ags_frequency_aliase_util_process_s64 (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of signed 64 bit data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_float ()

```
void
ags_frequency_aliase_util_process_float
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of single precision floating point data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_double ()

```
void
ags_frequency_aliase_util_process_double
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of double precision floating point data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process_complex ()

```
void
ags_frequency_aliase_util_process_complex
                                (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* of complex floating point data.

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_process ()

```
void
ags_frequency_aliase_util_process (AgsFrequencyAliaseUtil *frequency_aliase_util);
```

Process phase shift of *frequency_aliase_util* .

Parameters

frequency_aliase_util	the AgsFrequencyAliaseUtil

Since: 4.0.0

ags_frequency_aliase_util_get_type ()

```
GType
ags_frequency_aliase_util_get_type (void);
```

Types and Values

AGS_FREQUENCY_ALIASE_UTIL_DEFAULT_FORMAT

```
#define AGS_FREQUENCY_ALIASE_UTIL_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_TYPE_FREQUENCY_ALIASE_UTIL

```
#define AGS_TYPE_FREQUENCY_ALIASE_UTIL          (ags_frequency_aliase_util_get_type())
```

struct AgsFrequencyAliaseUtil

```
struct AgsFrequencyAliaseUtil {
  gpointer source;
  guint source_stride;

  gpointer phase_shifted_source;
  guint phase_shifted_source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;
};
```

5.14 AgsAudioBufferUtil

AgsAudioBufferUtil — audio buffer util

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_BUFFER_S8()
#define	AGS_AUDIO_BUFFER_S16()
#define	AGS_AUDIO_BUFFER_S24()
#define	AGS_AUDIO_BUFFER_S32()
#define	AGS_AUDIO_BUFFER_S64()
#define	AGS_AUDIO_BUFFER_FLOAT()
#define	AGS_AUDIO_BUFFER_DOUBLE()
#define	AGS_AUDIO_BUFFER_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_S8_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_S16_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_S24_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_S32_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_S64_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_FLOAT_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_DOUBLE_TO_COMPLEX()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S8()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S16()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S24()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S32()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S64()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_FLOAT()
#define	AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_DOUBLE()
guint	ags_audio_buffer_util_format_from_soundcard ()
guint	ags_audio_buffer_util_get_copy_mode ()
void	ags_audio_buffer_util_clear_float ()
void	ags_audio_buffer_util_clear_float32 ()
void	ags_audio_buffer_util_clear_double ()
void	ags_audio_buffer_util_clear_complex ()
void	ags_audio_buffer_util_clear_buffer ()
void	ags_audio_buffer_util_pong_s8 ()
void	ags_audio_buffer_util_pong_s16 ()
void	ags_audio_buffer_util_pong_s24 ()
void	ags_audio_buffer_util_pong_s32 ()
void	ags_audio_buffer_util_pong_s64 ()
void	ags_audio_buffer_util_pong_float ()
void	ags_audio_buffer_util_pong_double ()
void	ags_audio_buffer_util_pong_complex ()
void	ags_audio_buffer_util_pong ()
void	ags_audio_buffer_util_copy_s8_to_s8 ()
void	ags_audio_buffer_util_copy_s8_to_s16 ()
void	ags_audio_buffer_util_copy_s8_to_s24 ()
void	ags_audio_buffer_util_copy_s8_to_s32 ()
void	ags_audio_buffer_util_copy_s8_to_s64 ()
void	ags_audio_buffer_util_copy_s8_to_float ()

void	ags_audio_buffer_util_copy_s8_to_double ()
void	ags_audio_buffer_util_copy_s16_to_s8 ()
void	ags_audio_buffer_util_copy_s16_to_s16 ()
void	ags_audio_buffer_util_copy_s16_to_s24 ()
void	ags_audio_buffer_util_copy_s16_to_s32 ()
void	ags_audio_buffer_util_copy_s16_to_s64 ()
void	ags_audio_buffer_util_copy_s16_to_float ()
void	ags_audio_buffer_util_copy_s16_to_double ()
void	ags_audio_buffer_util_copy_s24_to_s8 ()
void	ags_audio_buffer_util_copy_s24_to_s16 ()
void	ags_audio_buffer_util_copy_s24_to_s24 ()
void	ags_audio_buffer_util_copy_s24_to_s32 ()
void	ags_audio_buffer_util_copy_s24_to_s64 ()
void	ags_audio_buffer_util_copy_s24_to_float ()
void	ags_audio_buffer_util_copy_s24_to_double ()
void	ags_audio_buffer_util_copy_s32_to_s8 ()
void	ags_audio_buffer_util_copy_s32_to_s16 ()
void	ags_audio_buffer_util_copy_s32_to_s24 ()
void	ags_audio_buffer_util_copy_s32_to_s32 ()
void	ags_audio_buffer_util_copy_s32_to_s64 ()
void	ags_audio_buffer_util_copy_s32_to_float ()
void	ags_audio_buffer_util_copy_s32_to_double ()
void	ags_audio_buffer_util_copy_s64_to_s8 ()
void	ags_audio_buffer_util_copy_s64_to_s16 ()
void	ags_audio_buffer_util_copy_s64_to_s24 ()
void	ags_audio_buffer_util_copy_s64_to_s32 ()
void	ags_audio_buffer_util_copy_s64_to_s64 ()
void	ags_audio_buffer_util_copy_s64_to_float ()
void	ags_audio_buffer_util_copy_s64_to_double ()
void	ags_audio_buffer_util_copy_float_to_s8 ()
void	ags_audio_buffer_util_copy_float_to_s16 ()
void	ags_audio_buffer_util_copy_float_to_s24 ()
void	ags_audio_buffer_util_copy_float_to_s32 ()
void	ags_audio_buffer_util_copy_float_to_s64 ()
void	ags_audio_buffer_util_copy_float_to_float ()
void	ags_audio_buffer_util_copy_float_to_double ()
void	ags_audio_buffer_util_copy_double_to_s8 ()
void	ags_audio_buffer_util_copy_double_to_s16 ()
void	ags_audio_buffer_util_copy_double_to_s24 ()
void	ags_audio_buffer_util_copy_double_to_s32 ()
void	ags_audio_buffer_util_copy_double_to_s64 ()
void	ags_audio_buffer_util_copy_double_to_float ()
void	ags_audio_buffer_util_copy_double_to_double ()
void	ags_audio_buffer_util_copy_s8_to_float32 ()
void	ags_audio_buffer_util_copy_s16_to_float32 ()
void	ags_audio_buffer_util_copy_s24_to_float32 ()
void	ags_audio_buffer_util_copy_s32_to_float32 ()
void	ags_audio_buffer_util_copy_s64_to_float32 ()
void	ags_audio_buffer_util_copy_float_to_float32 ()
void	ags_audio_buffer_util_copy_double_to_float32 ()
void	ags_audio_buffer_util_copy_float32_to_float32 ()
void	ags_audio_buffer_util_copy_float32_to_s8 ()
void	ags_audio_buffer_util_copy_float32_to_s16 ()
void	ags_audio_buffer_util_copy_float32_to_s24 ()
void	ags_audio_buffer_util_copy_float32_to_s32 ()
void	ags_audio_buffer_util_copy_float32_to_s64 ()

void	ags_audio_buffer_util_copy_float32_to_float ()
void	ags_audio_buffer_util_copy_float32_to_double ()
void	ags_audio_buffer_util_copy_s8_to_complex ()
void	ags_audio_buffer_util_copy_s16_to_complex ()
void	ags_audio_buffer_util_copy_s24_to_complex ()
void	ags_audio_buffer_util_copy_s32_to_complex ()
void	ags_audio_buffer_util_copy_s64_to_complex ()
void	ags_audio_buffer_util_copy_float_to_complex ()
void	ags_audio_buffer_util_copy_double_to_complex ()
void	ags_audio_buffer_util_copy_float32_to_complex ()
void	ags_audio_buffer_util_copy_complex_to_complex ()
void	ags_audio_buffer_util_copy_complex_to_s8 ()
void	ags_audio_buffer_util_copy_complex_to_s16 ()
void	ags_audio_buffer_util_copy_complex_to_s24 ()
void	ags_audio_buffer_util_copy_complex_to_s32 ()
void	ags_audio_buffer_util_copy_complex_to_s64 ()
void	ags_audio_buffer_util_copy_complex_to_float ()
void	ags_audio_buffer_util_copy_complex_to_double ()
void	ags_audio_buffer_util_copy_complex_to_float32 ()
void	ags_audio_buffer_util_copy_buffer_to_buffer ()
GType	ags_audio_buffer_util_get_type ()

Types and Values

enum	AgsAudioBufferUtilFormat
enum	AgsAudioBufferUtilSamplerate
enum	AgsAudioBufferUtilCopyMode
#define	AGS_TYPE_AUDIO_BUFFER_UTIL
struct	AgsAudioBufferUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsAudioBufferUtil
```

Includes

```
#include <ags/audio/ags_audio_buffer_util.h>
```

Description

Utility functions to process audio buffer.

Functions

AGS_AUDIO_BUFFER_S8()

```
#define AGS_AUDIO_BUFFER_S8(ptr) ((gint8 *)(ptr))
```

AGS_AUDIO_BUFFER_S16()

```
#define AGS_AUDIO_BUFFER_S16(ptr) ((gint16 *)(ptr))
```

AGS_AUDIO_BUFFER_S24()

```
#define AGS_AUDIO_BUFFER_S24(ptr) ((gint32 *)(ptr))
```

AGS_AUDIO_BUFFER_S32()

```
#define AGS_AUDIO_BUFFER_S32(ptr) ((gint32 *)(ptr))
```

AGS_AUDIO_BUFFER_S64()

```
#define AGS_AUDIO_BUFFER_S64(ptr) ((gint64 *)(ptr))
```

AGS_AUDIO_BUFFER_FLOAT()

```
#define AGS_AUDIO_BUFFER_FLOAT(ptr) ((gfloat *)(ptr))
```

AGS_AUDIO_BUFFER_DOUBLE()

```
#define AGS_AUDIO_BUFFER_DOUBLE(ptr) ((gdouble *)(ptr))
```

AGS_AUDIO_BUFFER_COMPLEX()

```
#define AGS_AUDIO_BUFFER_COMPLEX(ptr) ((AgsComplex *)(ptr))
```

AGS_AUDIO_BUFFER_UTIL_S8_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_S8_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_S16_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_S16_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_S24_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_S24_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_S32_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_S32_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_S64_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_S64_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_FLOAT_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_FLOAT_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_DOUBLE_TO_COMPLEX()

```
#define             AGS_AUDIO_BUFFER_UTIL_DOUBLE_TO_COMPLEX(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S8()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S8(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S16()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S16(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S24()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S24(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S32()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S32(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S64()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S64(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_FLOAT()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_FLOAT(x_val, x_retval)
```

AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_DOUBLE()

```
#define             AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_DOUBLE(x_val, x_retval)
```

ags_audio_buffer_util_format_from_soundcard ()

```
guint
ags_audio_buffer_util_format_from_soundcard
                                (guint soundcard_format);
```

Translate soundcard format to audio buffer util format.

Parameters

soundcard_format	the soundcard bit mode	

Returns

the converted format

Since: 3.0.0

ags_audio_buffer_util_get_copy_mode ()

```
guint
ags_audio_buffer_util_get_copy_mode (guint destination_format,
                                     guint source_format);
```

Get copy mode for given destination and source format.

Parameters

destination_format	the destination buffer format	
source_format	the source buffer format	

Returns

the desired copy mode

Since: 3.0.0

ags_audio_buffer_util_clear_float ()

```
void
ags_audio_buffer_util_clear_float (gfloat *buffer,
                                   guint channels,
                                   guint count);
```

Clears a buffer.

Parameters

buffer	the buffer to clear	
channels	number of channels	
count	count frames	

Since: 3.0.0

ags_audio_buffer_util_clear_float32 ()

```
void
ags_audio_buffer_util_clear_float32 (Float32 *buffer,
                                     guint channels,
                                     guint count);
```

Clears a buffer.

Parameters

buffer	the buffer to clear	
channels	number of channels	
count	count frames	

Since: 3.0.0

ags_audio_buffer_util_clear_double ()

```
void
ags_audio_buffer_util_clear_double (gdouble *buffer,
                                    guint channels,
                                    guint count);
```

Clears a buffer.

Parameters

buffer	the buffer to clear	
channels	number of channels	
count	count frames	

Since: 3.0.0

ags_audio_buffer_util_clear_complex ()

```
void
ags_audio_buffer_util_clear_complex (AgsComplex *buffer,
                                     guint channels,
                                     guint count);
```

Clears a buffer.

Parameters

buffer	the buffer to clear	
channels	number of channels	
count	count frames	

Since: 3.0.0

ags_audio_buffer_util_clear_buffer ()

```
void
ags_audio_buffer_util_clear_buffer (void *buffer,
                                    guint channels,
                                    guint count,
                                    guint format);
```

Clears the audio data.

Parameters

buffer	the audio data	
channels	the number of channels	
count	number of frames	
format	the format	

Since: 3.0.0

ags_audio_buffer_util_pong_s8 ()

```
void
ags_audio_buffer_util_pong_s8 (gint8 *destination,
                               guint dchannels,
                               gint8 *source,
                               guint schannels,
                               guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_s16 ()

```
void
ags_audio_buffer_util_pong_s16 (gint16 *destination,
                                guint dchannels,
                                gint16 *source,
                                guint schannels,
                                guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_s24 ()

```
void
ags_audio_buffer_util_pong_s24 (gint32 *destination,
                                guint dchannels,
                                gint32 *source,
                                guint schannels,
                                guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_s32 ()

```
void
ags_audio_buffer_util_pong_s32 (gint32 *destination,
                                guint dchannels,
                                gint32 *source,
                                guint schannels,
                                guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_s64 ()

```
void
ags_audio_buffer_util_pong_s64 (gint64 *destination,
                                guint dchannels,
                                gint64 *source,
                                guint schannels,
                                guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_float ()

```
void
ags_audio_buffer_util_pong_float (gfloat *destination,
                                  guint dchannels,
                                  gfloat *source,
                                  guint schannels,
                                  guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_double ()

```
void
ags_audio_buffer_util_pong_double (gdouble *destination,
                                   guint dchannels,
                                   gdouble *source,
                                   guint schannels,
                                   guint count);
```

Pingpong copy *source* to *destination* .

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong_complex ()

```
void
ags_audio_buffer_util_pong_complex (AgsComplex *destination,
                                    guint dchannels,
                                    AgsComplex *source,
                                    guint schannels,
                                    guint count);
```

Pingpong copy *source* to *destination*.

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	

Since: 3.4.0

ags_audio_buffer_util_pong ()

```
void
ags_audio_buffer_util_pong (void *destination,
                            guint dchannels,
                            void *source,
                            guint schannels,
                            guint count,
                            guint format);
```

Pingpong copy *source* to *destination*.

Parameters

destination	the destination	
dchannels	the destination channels	
source	the source	
schannels	the source channels	
count	the count of frames	
format	the format	

Since: 3.4.0

ags_audio_buffer_util_copy_s8_to_s8 ()

```
void
ags_audio_buffer_util_copy_s8_to_s8 (gint8 *destination,
                                     guint dchannels,
                                     gint8 *source,
                                     guint schannels,
                                     guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_s16 ()

```
void
ags_audio_buffer_util_copy_s8_to_s16 (gint16 *destination,
                                      guint dchannels,
                                      gint8 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_s24 ()

```
void
ags_audio_buffer_util_copy_s8_to_s24 (gint32 *destination,
                                      guint dchannels,
                                      gint8 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_s32 ()

```
void
ags_audio_buffer_util_copy_s8_to_s32 (gint32 *destination,
                                      guint dchannels,
                                      gint8 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_s64 ()

```
void
ags_audio_buffer_util_copy_s8_to_s64 (gint64 *destination,
                                      guint dchannels,
                                      gint8 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	

schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_float ()

```
void
ags_audio_buffer_util_copy_s8_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gint8 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_double ()

```
void
ags_audio_buffer_util_copy_s8_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gint8 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_s8 ()

```
void
ags_audio_buffer_util_copy_s16_to_s8 (gint8 *destination,
                                      guint dchannels,
                                      gint16 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_s16 ()

```
void
ags_audio_buffer_util_copy_s16_to_s16 (gint16 *destination,
                                       guint dchannels,
                                       gint16 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_s24 ()

```
void
ags_audio_buffer_util_copy_s16_to_s24 (gint32 *destination,
                                       guint dchannels,
                                       gint16 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_s32 ()

```
void
ags_audio_buffer_util_copy_s16_to_s32 (gint32 *destination,
                                       guint dchannels,
                                       gint16 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_s64 ()

```
void
ags_audio_buffer_util_copy_s16_to_s64 (gint64 *destination,
                                       guint dchannels,
                                       gint16 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	

schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_float ()

```
void
ags_audio_buffer_util_copy_s16_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gint16 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_double ()

```
void
ags_audio_buffer_util_copy_s16_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gint16 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_s8 ()

```
void
ags_audio_buffer_util_copy_s24_to_s8 (gint8 *destination,
                                      guint dchannels,
                                      gint32 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_s16 ()

```
void
ags_audio_buffer_util_copy_s24_to_s16 (gint16 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_s24 ()

```
void
ags_audio_buffer_util_copy_s24_to_s24 (gint32 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_s32 ()

```
void
ags_audio_buffer_util_copy_s24_to_s32 (gint32 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_s64 ()

```
void
ags_audio_buffer_util_copy_s24_to_s64 (gint64 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	

schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_float ()

```
void
ags_audio_buffer_util_copy_s24_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_double ()

```
void
ags_audio_buffer_util_copy_s24_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_s8 ()

```
void
ags_audio_buffer_util_copy_s32_to_s8 (gint8 *destination,
                                      guint dchannels,
                                      gint32 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_s16 ()

```
void
ags_audio_buffer_util_copy_s32_to_s16 (gint16 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_s24 ()

```
void
ags_audio_buffer_util_copy_s32_to_s24 (gint32 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_s32 ()

```
void
ags_audio_buffer_util_copy_s32_to_s32 (gint32 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_s64 ()

```
void
ags_audio_buffer_util_copy_s32_to_s64 (gint64 *destination,
                                       guint dchannels,
                                       gint32 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	

schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_float ()

```
void
ags_audio_buffer_util_copy_s32_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_double ()

```
void
ags_audio_buffer_util_copy_s32_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_s8 ()

```
void
ags_audio_buffer_util_copy_s64_to_s8 (gint8 *destination,
                                      guint dchannels,
                                      gint64 *source,
                                      guint schannels,
                                      guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_s16 ()

```
void
ags_audio_buffer_util_copy_s64_to_s16 (gint16 *destination,
                                       guint dchannels,
                                       gint64 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_s24 ()

```
void
ags_audio_buffer_util_copy_s64_to_s24 (gint32 *destination,
                                       guint dchannels,
                                       gint64 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_s32 ()

```
void
ags_audio_buffer_util_copy_s64_to_s32 (gint32 *destination,
                                       guint dchannels,
                                       gint64 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_s64 ()

```
void
ags_audio_buffer_util_copy_s64_to_s64 (gint64 *destination,
                                       guint dchannels,
                                       gint64 *source,
                                       guint schannels,
                                       guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	

schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_float ()

```
void
ags_audio_buffer_util_copy_s64_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gint64 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_double ()

```
void
ags_audio_buffer_util_copy_s64_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gint64 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_s8 ()

```
void
ags_audio_buffer_util_copy_float_to_s8
                                (gint8 *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_s16 ()

```
void
ags_audio_buffer_util_copy_float_to_s16
                                (gint16 *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_s24 ()

```
void
ags_audio_buffer_util_copy_float_to_s24
                                (gint32 *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_s32 ()

```
void
ags_audio_buffer_util_copy_float_to_s32
                                (gint32 *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_s64 ()

```
void
ags_audio_buffer_util_copy_float_to_s64
                                (gint64 *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_float ()

```
void
ags_audio_buffer_util_copy_float_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_double ()

```
void
ags_audio_buffer_util_copy_float_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_s8 ()

```
void
ags_audio_buffer_util_copy_double_to_s8
                                (gint8 *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_s16 ()

```
void
ags_audio_buffer_util_copy_double_to_s16
                                (gint16 *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	

dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_s24 ()

```
void
ags_audio_buffer_util_copy_double_to_s24
                                (gint32 *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_s32 ()

```
void
ags_audio_buffer_util_copy_double_to_s32
                                (gint32 *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	

count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_s64 ()

```
void
ags_audio_buffer_util_copy_double_to_s64
                                (gint64 *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_float ()

```
void
ags_audio_buffer_util_copy_double_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_double ()

```
void
ags_audio_buffer_util_copy_double_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_float32 ()

```
void
ags_audio_buffer_util_copy_s8_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 gint8 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_float32 ()

```
void
ags_audio_buffer_util_copy_s16_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 gint16 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_float32 ()

```
void
ags_audio_buffer_util_copy_s24_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_float32 ()

```
void
ags_audio_buffer_util_copy_s32_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_float32 ()

```
void
ags_audio_buffer_util_copy_s64_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 gint64 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_float32 ()

```
void
ags_audio_buffer_util_copy_float_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 float *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_float32 ()

```
void
ags_audio_buffer_util_copy_double_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_float32 ()

```
void
ags_audio_buffer_util_copy_float32_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	

dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_s8 ()

```
void
ags_audio_buffer_util_copy_float32_to_s8
                                (gint8 *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_s16 ()

```
void
ags_audio_buffer_util_copy_float32_to_s16
                                (gint16 *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	

count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_s24 ()

```
void
ags_audio_buffer_util_copy_float32_to_s24
                                (gint32 *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_s32 ()

```
void
ags_audio_buffer_util_copy_float32_to_s32
                                (gint32 *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_s64 ()

```
void
ags_audio_buffer_util_copy_float32_to_s64
                                (gint64 *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_float ()

```
void
ags_audio_buffer_util_copy_float32_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_double ()

```
void
ags_audio_buffer_util_copy_float32_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s8_to_complex ()

```
void
ags_audio_buffer_util_copy_s8_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gint8 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s16_to_complex ()

```
void
ags_audio_buffer_util_copy_s16_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gint16 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s24_to_complex ()

```
void
ags_audio_buffer_util_copy_s24_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s32_to_complex ()

```
void
ags_audio_buffer_util_copy_s32_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gint32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_s64_to_complex ()

```
void
ags_audio_buffer_util_copy_s64_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gint64 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float_to_complex ()

```
void
ags_audio_buffer_util_copy_float_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gfloat *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	

dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_double_to_complex ()

```
void
ags_audio_buffer_util_copy_double_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 gdouble *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_float32_to_complex ()

```
void
ags_audio_buffer_util_copy_float32_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 Float32 *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	

count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_complex ()

```
void
ags_audio_buffer_util_copy_complex_to_complex
                                (AgsComplex *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_s8 ()

```
void
ags_audio_buffer_util_copy_complex_to_s8
                                (gint8 *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_s16 ()

```
void
ags_audio_buffer_util_copy_complex_to_s16
                                (gint16 *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_s24 ()

```
void
ags_audio_buffer_util_copy_complex_to_s24
                                (gint32 *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_s32 ()

```
void
ags_audio_buffer_util_copy_complex_to_s32
                                (gint32 *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_s64 ()

```
void
ags_audio_buffer_util_copy_complex_to_s64
                                (gint64 *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_float ()

```
void
ags_audio_buffer_util_copy_complex_to_float
                                (gfloat *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_double ()

```
void
ags_audio_buffer_util_copy_complex_to_double
                                (gdouble *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_complex_to_float32 ()

```
void
ags_audio_buffer_util_copy_complex_to_float32
                                (Float32 *destination,
                                 guint dchannels,
                                 AgsComplex *source,
                                 guint schannels,
                                 guint count);
```

Copy audio data using additive strategy.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
source	source buffer	
schannels	source buffer's count of channels	
count	number of frames to copy	

Since: 3.0.0

ags_audio_buffer_util_copy_buffer_to_buffer ()

```
void
ags_audio_buffer_util_copy_buffer_to_buffer
                                (void *destination,
                                 guint dchannels,
                                 guint doffset,
                                 void *source,
                                 guint schannels,
                                 guint soffset,
                                 guint count,
                                 guint mode);
```

Wrapper function to copy functions. Doing type conversion.

Parameters

destination	destination buffer	
dchannels	destination buffer's count of channels	
doffset	start frame of destination	
source	source buffer	
schannels	source buffer's count of channels	
soffset	start frame of source	
count	number of frames to copy	
mode	specified type conversion as described	

Since: 3.0.0

ags_audio_buffer_util_get_type ()

```
GType
ags_audio_buffer_util_get_type (void);
```

Types and Values

enum AgsAudioBufferUtilFormat

AgsAudioBufferUtilFormat specifies the audio data representation.

Members

AGS_AUDIO_BUFFER_UTIL_S8	signed 8 bit raw pcm
AGS_AUDIO_BUFFER_UTIL_S16	signed 16 bit raw pcm
AGS_AUDIO_BUFFER_UTIL_S24	signed 24 bit raw pcm
AGS_AUDIO_BUFFER_UTIL_S32	signed 32 bit raw pcm
AGS_AUDIO_BUFFER_UTIL_S64	signed 64 bit raw pcm
AGS_AUDIO_BUFFER_UTIL_FLOAT	signed float raw pcm
AGS_AUDIO_BUFFER_UTIL_DOUBLE	signed dou- ble raw pcm
AGS_AUDIO_BUFFER_UTIL_FLOAT32	signed float32 raw pcm
AGS_AUDIO_BUFFER_UTIL_COMPLEX	complex data type

enum AgsAudioBufferUtilSamplerate

Some common samplerates in Hertz.

Members

AGS_AUDIO_BUFFER_UTIL_8000HZ	8000 Hz
AGS_AUDIO_BUFFER_UTIL_44100HZ	44100 Hz

AGS_AUDIO_BUFFER_UTIL_48000HZ	48000 Hz
AGS_AUDIO_BUFFER_UTIL_96000HZ	96000 Hz
AGS_AUDIO_BUFFER_UTIL_176400HZ	176400 Hz
AGS_AUDIO_BUFFER_UTIL_192000HZ	192000 Hz
AGS_AUDIO_BUFFER_UTIL_352800HZ	352800 Hz
AGS_AUDIO_BUFFER_UTIL_2822400HZ	2822400 Hz
AGS_AUDIO_BUFFER_UTIL_5644800HZ	5644800 Hz

enum AgsAudioBufferUtilCopyMode

Copy modes.

Members

AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_S8
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_S8

AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_S8
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_S8

AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_S8
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_S8

AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_S8
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_FLOAT32

AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_FLOAT32
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_FLOAT32
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_FLOAT32
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_FLOAT32
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_FLOAT32
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_S8
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_S64
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_DOUBLE

AGS_AUDIO_BUFFER_UTIL_COPY_S8_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_S8 to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_S16_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_S16 to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_S24_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_S24 to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_S32_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_S32 to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_S64_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_S64 to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_FLOAT to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_DOUBLE_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_DOUBLE to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_FLOAT32_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_FLOAT32 to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_COMPLEX	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_COMPLEX
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_S8	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_S8
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_S16	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_S16
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_S24	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_S24
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_S32	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_S32
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_S64	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_S64

AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_FLOAT	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_FLOAT
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_DOUBLE	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_DOUBLE
AGS_AUDIO_BUFFER_UTIL_COPY_COMPLEX_TO_FLOAT32	copy AGS_AUDIO_BUFFER_UTIL_COMPLEX to AGS_AUDIO_BUFFER_UTIL_FLOAT32

AGS_TYPE_AUDIO_BUFFER_UTIL

```
#define AGS_TYPE_AUDIO_BUFFER_UTIL          (ags_audio_buffer_util_get_type())
```

struct AgsAudioBufferUtil

```
struct AgsAudioBufferUtil {
  //empty
};
```

5.15 AgsVolumeUtil

AgsVolumeUtil — Boxed type of volume util

Stability Level

Stable, unless otherwise indicated

Functions

AgsVolumeUtil *	ags_volume_util_alloc ()
gpointer	ags_volume_util_copy ()
void	ags_volume_util_free ()
gpointer	ags_volume_util_get_destination ()
void	ags_volume_util_set_destination ()
guint	ags_volume_util_get_destination_stride ()
void	ags_volume_util_set_destination_stride ()
gpointer	ags_volume_util_get_source ()
void	ags_volume_util_set_source ()
guint	ags_volume_util_get_source_stride ()
void	ags_volume_util_set_source_stride ()
guint	ags_volume_util_get_buffer_length ()
void	ags_volume_util_set_buffer_length ()
guint	ags_volume_util_get_format ()
void	ags_volume_util_set_format ()
gdouble	ags_volume_util_get_volume ()
void	ags_volume_util_set_volume ()

void	ags_volume_util_compute_s8 ()
void	ags_volume_util_compute_s16 ()
void	ags_volume_util_compute_s24 ()
void	ags_volume_util_compute_s32 ()
void	ags_volume_util_compute_s64 ()
void	ags_volume_util_compute_float ()
void	ags_volume_util_compute_double ()
void	ags_volume_util_compute_complex ()
void	ags_volume_util_compute ()
#define	AGS_VOLUME_UTIL()
GType	ags_volume_util_get_type ()

Types and Values

#define	AGS_VOLUME_UTIL_DEFAULT_FORMAT
#define	AGS_TYPE_VOLUME_UTIL
struct	AgsVolumeUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsVolumeUtil
```

Includes

```
#include <ags/audio/ags_volume_util.h>
```

Description

Boxed type of volume util data type.

Functions

ags_volume_util_alloc ()

```
AgsVolumeUtil~*
ags_volume_util_alloc ();
```

Allocate AgsVolumeUtil

Returns

a new AgsVolumeUtil

Since: 3.9.2

ags_volume_util_copy ()

```
gpointer
ags_volume_util_copy (AgsVolumeUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsVolumeUtil

Returns

a pointer of the new AgsVolumeUtil

Since: 3.9.2

ags_volume_util_free ()

```
void
ags_volume_util_free (AgsVolumeUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_get_destination ()

```
gpointer
ags_volume_util_get_destination (AgsVolumeUtil *volume_util);
```

Get destination buffer of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil

Returns

the destination buffer

Since: 3.9.2

ags_volume_util_set_destination ()

```
void
ags_volume_util_set_destination (AgsVolumeUtil *volume_util,
                                 gpointer destination);
```

Set *destination* buffer of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	
destination	the destination buffer	

Since: 3.9.2

ags_volume_util_get_destination_stride ()

```
guint
ags_volume_util_get_destination_stride
                                (AgsVolumeUtil *volume_util);
```

Get destination stride of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	

Returns

the destination buffer stride

Since: 3.9.2

ags_volume_util_set_destination_stride ()

```
void
ags_volume_util_set_destination_stride
                                (AgsVolumeUtil *volume_util,
                                 guint destination_stride);
```

Set *destination* stride of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	
destination_stride	the destination buffer stride	

Since: 3.9.2

ags_volume_util_get_source ()

```
gpointer
ags_volume_util_get_source (AgsVolumeUtil *volume_util);
```

Get source buffer of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	

Returns

the source buffer

Since: 3.9.2

ags_volume_util_set_source ()

```
void
ags_volume_util_set_source (AgsVolumeUtil *volume_util,
                            gpointer source);
```

Set *source* buffer of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	
source	the source buffer	

Since: 3.9.2

ags_volume_util_get_source_stride ()

```
guint
ags_volume_util_get_source_stride (AgsVolumeUtil *volume_util);
```

Get source stride of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	

Returns

the source buffer stride

Since: 3.9.2

ags_volume_util_set_source_stride ()

```
void
ags_volume_util_set_source_stride (AgsVolumeUtil *volume_util,
                                   guint source_stride);
```

Set *source* stride of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	
source_stride	the source buffer stride	

Since: 3.9.2

ags_volume_util_get_buffer_length ()

```
guint
ags_volume_util_get_buffer_length (AgsVolumeUtil *volume_util);
```

Get buffer length of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil

Returns

the buffer length

Since: 3.9.2

ags_volume_util_set_buffer_length ()

```
void
ags_volume_util_set_buffer_length (AgsVolumeUtil *volume_util,
                                   guint buffer_length);
```

Set *buffer_length* of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil
buffer_length	the buffer length

Since: 3.9.2

ags_volume_util_get_format ()

```
guint
ags_volume_util_get_format (AgsVolumeUtil *volume_util);
```

Get format of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil

Returns

the format

Since: 3.9.6

ags_volume_util_set_format ()

```
void
ags_volume_util_set_format (AgsVolumeUtil *volume_util,
                            guint format);
```

Set *format* of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	
format	the format	

Since: 3.9.6

ags_volume_util_get_volume ()

```
gdouble
ags_volume_util_get_volume (AgsVolumeUtil *volume_util);
```

Get volume of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	

Returns

the volume

Since: 3.9.2

ags_volume_util_set_volume ()

```
void
ags_volume_util_set_volume (AgsVolumeUtil *volume_util,
                            gdouble volume);
```

Set *volume* of *volume_util* .

Parameters

volume_util	the AgsVolumeUtil	
volume	the volume	

Since: 3.9.2

ags_volume_util_compute_s8 ()

```
void
ags_volume_util_compute_s8 (AgsVolumeUtil *volume_util);
```

Compute volume of signed 8 bit data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_s16 ()

```
void
ags_volume_util_compute_s16 (AgsVolumeUtil *volume_util);
```

Compute volume of signed 16 bit data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_s24 ()

```
void
ags_volume_util_compute_s24 (AgsVolumeUtil *volume_util);
```

Compute volume of signed 24 bit data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_s32 ()

```
void
ags_volume_util_compute_s32 (AgsVolumeUtil *volume_util);
```

Compute volume of signed 32 bit data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_s64 ()

```
void
ags_volume_util_compute_s64 (AgsVolumeUtil *volume_util);
```

Compute volume of signed 64 bit data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_float ()

```
void
ags_volume_util_compute_float (AgsVolumeUtil *volume_util);
```

Compute volume of floating point data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_double ()

```
void
ags_volume_util_compute_double (AgsVolumeUtil *volume_util);
```

Compute volume of double floating point data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute_complex ()

```
void
ags_volume_util_compute_complex (AgsVolumeUtil *volume_util);
```

Compute volume of complex floating point data.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

ags_volume_util_compute ()

```
void
ags_volume_util_compute (AgsVolumeUtil *volume_util);
```

Compute volume.

Parameters

volume_util	the AgsVolumeUtil

Since: 3.9.2

AGS_VOLUME_UTIL()

```
#define AGS_VOLUME_UTIL(ptr) ((AgsVolumeUtil *)(ptr))
```

ags_volume_util_get_type ()

```
GType
ags_volume_util_get_type (void);
```

Types and Values

AGS_VOLUME_UTIL_DEFAULT_FORMAT

```
#define AGS_VOLUME_UTIL_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_TYPE_VOLUME_UTIL

```
#define AGS_TYPE_VOLUME_UTIL         (ags_volume_util_get_type())
```

struct AgsVolumeUtil

```
struct AgsVolumeUtil {
  gpointer destination;
  guint destination_stride;

  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;

  gdouble volume;
};
```

5.16 AgsEnvelopeUtil

AgsEnvelopeUtil — Boxed type of envelope util

Stability Level

Stable, unless otherwise indicated

Functions

AgsEnvelopeUtil *	ags_envelope_util_alloc ()
gpointer	ags_envelope_util_copy ()
void	ags_envelope_util_free ()
gpointer	ags_envelope_util_get_destination ()
void	ags_envelope_util_set_destination ()
guint	ags_envelope_util_get_destination_stride ()
void	ags_envelope_util_set_destination_stride ()
gpointer	ags_envelope_util_get_source ()
void	ags_envelope_util_set_source ()
guint	ags_envelope_util_get_source_stride ()
void	ags_envelope_util_set_source_stride ()
guint	ags_envelope_util_get_buffer_length ()
void	ags_envelope_util_set_buffer_length ()
guint	ags_envelope_util_get_format ()
void	ags_envelope_util_set_format ()
gdouble	ags_envelope_util_get_volume ()
void	ags_envelope_util_set_volume ()
gdouble	ags_envelope_util_get_amount ()
void	ags_envelope_util_set_amount ()
void	ags_envelope_util_compute_s8 ()
void	ags_envelope_util_compute_s16 ()
void	ags_envelope_util_compute_s24 ()
void	ags_envelope_util_compute_s32 ()
void	ags_envelope_util_compute_s64 ()
void	ags_envelope_util_compute_float ()
void	ags_envelope_util_compute_double ()
void	ags_envelope_util_compute_complex ()
void	ags_envelope_util_compute ()
#define	AGS_ENVELOPE_UTIL()
GType	ags_envelope_util_get_type ()

Types and Values

#define	AGS_ENVELOPE_UTIL_DEFAULT_FORMAT
#define	AGS_ENVELOPE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT
#define	AGS_TYPE_ENVELOPE_UTIL
struct	AgsEnvelopeUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsEnvelopeUtil
```

Includes

```
#include <ags/audio/ags_envelope_util.h>
```

Description

Boxed type of envelope util data type.

Functions

ags_envelope_util_alloc ()

```
AgsEnvelopeUtil~*
ags_envelope_util_alloc ();
```

Allocate AgsEnvelopeUtil

Returns

a new AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_copy ()

```
gpointer
ags_envelope_util_copy (AgsEnvelopeUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsEnvelopeUtil

Returns

a pointer of the new AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_free ()

```
void
ags_envelope_util_free (AgsEnvelopeUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_get_destination ()

```
gpointer
ags_envelope_util_get_destination (AgsEnvelopeUtil *envelope_util);
```

Get destination buffer of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	

Returns

the destination buffer

Since: 3.9.2

ags_envelope_util_set_destination ()

```
void
ags_envelope_util_set_destination (AgsEnvelopeUtil *envelope_util,
                                   gpointer destination);
```

Set *destination* buffer of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
destination	the destination buffer	

Since: 3.9.2

ags_envelope_util_get_destination_stride ()

```
guint
ags_envelope_util_get_destination_stride
                                (AgsEnvelopeUtil *envelope_util);
```

Get destination stride of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	

Returns

the destination buffer stride

Since: 3.9.2

ags_envelope_util_set_destination_stride ()

```
void
ags_envelope_util_set_destination_stride
                                (AgsEnvelopeUtil *envelope_util,
                                 guint destination_stride);
```

Set *destination* stride of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
destination_stride	the destination buffer stride	

Since: 3.9.2

ags_envelope_util_get_source ()

```
gpointer
ags_envelope_util_get_source (AgsEnvelopeUtil *envelope_util);
```

Get source buffer of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	

Returns

the source buffer

Since: 3.9.2

ags_envelope_util_set_source ()

```
void
ags_envelope_util_set_source (AgsEnvelopeUtil *envelope_util,
                              gpointer source);
```

Set *source* buffer of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
source	the source buffer	

Since: 3.9.2

ags_envelope_util_get_source_stride ()

```
guint
ags_envelope_util_get_source_stride (AgsEnvelopeUtil *envelope_util);
```

Get source stride of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	

Returns

the source buffer stride

Since: 3.9.2

ags_envelope_util_set_source_stride ()

```
void
ags_envelope_util_set_source_stride (AgsEnvelopeUtil *envelope_util,
                                     guint source_stride);
```

Set *source* stride of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
source_stride	the source buffer stride	

Since: 3.9.2

ags_envelope_util_get_buffer_length ()

```
guint
ags_envelope_util_get_buffer_length (AgsEnvelopeUtil *envelope_util);
```

Get buffer length of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	

Returns

the buffer length

Since: 3.9.2

ags_envelope_util_set_buffer_length ()

```
void
ags_envelope_util_set_buffer_length (AgsEnvelopeUtil *envelope_util,
                                     guint buffer_length);
```

Set *buffer_length* of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
buffer_length	the buffer length	

Since: 3.9.2

ags_envelope_util_get_format ()

```
guint
ags_envelope_util_get_format (AgsEnvelopeUtil *envelope_util);
```

Get format of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil

Returns

the format

Since: 3.9.6

ags_envelope_util_set_format ()

```
void
ags_envelope_util_set_format (AgsEnvelopeUtil *envelope_util,
                              guint format);
```

Set *format* of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil
format	the format

Since: 3.9.6

ags_envelope_util_get_volume ()

```
gdouble
ags_envelope_util_get_volume (AgsEnvelopeUtil *envelope_util);
```

Get volume of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil

Returns

the volume

Since: 3.9.2

ags_envelope_util_set_volume ()

```
void
ags_envelope_util_set_volume (AgsEnvelopeUtil *envelope_util,
                              gdouble volume);
```

Set *volume* of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
volume	the volume	

Since: 3.9.2

ags_envelope_util_get_amount ()

```
gdouble
ags_envelope_util_get_amount (AgsEnvelopeUtil *envelope_util);
```

Get amount of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	

Returns

the amount

Since: 3.9.2

ags_envelope_util_set_amount ()

```
void
ags_envelope_util_set_amount (AgsEnvelopeUtil *envelope_util,
                              gdouble amount);
```

Set *amount* of *envelope_util* .

Parameters

envelope_util	the AgsEnvelopeUtil	
amount	the amount	

Since: 3.9.2

ags_envelope_util_compute_s8 ()

```
void
ags_envelope_util_compute_s8 (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of signed 8 bit data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_s16 ()

```
void
ags_envelope_util_compute_s16 (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of signed 16 bit data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_s24 ()

```
void
ags_envelope_util_compute_s24 (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of signed 24 bit data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_s32 ()

```
void
ags_envelope_util_compute_s32 (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of signed 32 bit data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_s64 ()

```
void
ags_envelope_util_compute_s64 (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of signed 64 bit data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_float ()

```
void
ags_envelope_util_compute_float (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of floating point data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_double ()

```
void
ags_envelope_util_compute_double (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of double floating point data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute_complex ()

```
void
ags_envelope_util_compute_complex (AgsEnvelopeUtil *envelope_util);
```

Compute envelope of complex floating point data.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

ags_envelope_util_compute ()

```
void
ags_envelope_util_compute (AgsEnvelopeUtil *envelope_util);
```

Compute envelope.

Parameters

envelope_util	the AgsEnvelopeUtil

Since: 3.9.2

AGS_ENVELOPE_UTIL()

```
#define AGS_ENVELOPE_UTIL(ptr) ((AgsEnvelopeUtil *)(ptr))
```

ags_envelope_util_get_type ()

```
GType
ags_envelope_util_get_type (void);
```

Types and Values

AGS_ENVELOPE_UTIL_DEFAULT_FORMAT

```
#define AGS_ENVELOPE_UTIL_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_ENVELOPE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT

```
#define AGS_ENVELOPE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT (AGS_AUDIO_BUFFER_UTIL_S16)
```

AGS_TYPE_ENVELOPE_UTIL

```
#define AGS_TYPE_ENVELOPE_UTIL         (ags_envelope_util_get_type())
```

struct AgsEnvelopeUtil

```
struct AgsEnvelopeUtil {
  gpointer destination;
  guint destination_stride;

  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;

  gdouble volume;
  gdouble amount;
};
```

5.17 AgsPeakUtil

AgsPeakUtil — Boxed type of peak util

Stability Level

Stable, unless otherwise indicated

Functions

AgsPeakUtil *	ags_peak_util_alloc ()
gpointer	ags_peak_util_copy ()
void	ags_peak_util_free ()
gpointer	ags_peak_util_get_source ()
void	ags_peak_util_set_source ()
guint	ags_peak_util_get_source_stride ()
void	ags_peak_util_set_source_stride ()
guint	ags_peak_util_get_buffer_length ()
void	ags_peak_util_set_buffer_length ()
guint	ags_peak_util_get_format ()
void	ags_peak_util_set_format ()
guint	ags_peak_util_get_samplerate ()
void	ags_peak_util_set_samplerate ()
gdouble	ags_peak_util_get_harmonic_rate ()
void	ags_peak_util_set_harmonic_rate ()
gdouble	ags_peak_util_get_pressure_factor ()
void	ags_peak_util_set_pressure_factor ()
gdouble	ags_peak_util_get_peak ()
void	ags_peak_util_set_peak ()
void	ags_peak_util_compute_s8 ()
void	ags_peak_util_compute_s16 ()
void	ags_peak_util_compute_s24 ()
void	ags_peak_util_compute_s32 ()
void	ags_peak_util_compute_s64 ()
void	ags_peak_util_compute_float ()
void	ags_peak_util_compute_double ()
void	ags_peak_util_compute_complex ()
void	ags_peak_util_compute ()
#define	AGS_PEAK_UTIL()
GType	ags_peak_util_get_type ()

Types and Values

#define	AGS_PEAK_UTIL_DEFAULT_FORMAT
#define	AGS_PEAK_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT
#define	AGS_PEAK_UTIL_DEFAULT_SAMPLERATE
#define	AGS_PEAK_UTIL_DEFAULT_HARMONIC_RATE
#define	AGS_PEAK_UTIL_DEFAULT_PRESSURE_FACTOR
#define	AGS_TYPE_PEAK_UTIL
struct	AgsPeakUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsPeakUtil
```

Includes

```
#include <ags/audio/ags_peak_util.h>
```

Description

Boxed type of peak util data type.

Functions

ags_peak_util_alloc ()

```
AgsPeakUtil~*
ags_peak_util_alloc ();
```

Allocate AgsPeakUtil

Returns

a new AgsPeakUtil

Since: 3.9.2

ags_peak_util_copy ()

```
gpointer
ags_peak_util_copy (AgsPeakUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsPeakUtil

Returns

a pointer of the new AgsPeakUtil

Since: 3.9.2

ags_peak_util_free ()

```
void
ags_peak_util_free (AgsPeakUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_get_source ()

```
gpointer
ags_peak_util_get_source (AgsPeakUtil *peak_util);
```

Get source buffer of *peak_util* .

Parameters

peak_util	the AgsPeakUtil

Returns

the source buffer

Since: 3.9.2

ags_peak_util_set_source ()

```
void
ags_peak_util_set_source (AgsPeakUtil *peak_util,
                          gpointer source);
```

Set *source* buffer of *peak_util* .

Parameters

peak_util	the AgsPeakUtil
source	the source buffer

Since: 3.9.2

ags_peak_util_get_source_stride ()

```
guint
ags_peak_util_get_source_stride (AgsPeakUtil *peak_util);
```

Get source stride of *peak_util* .

Parameters

peak_util	the AgsPeakUtil

Returns

the source buffer stride

Since: 3.9.2

ags_peak_util_set_source_stride ()

```
void
ags_peak_util_set_source_stride (AgsPeakUtil *peak_util,
                                 guint source_stride);
```

Set *source* stride of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	
source_stride	the source buffer stride	

Since: 3.9.2

ags_peak_util_get_buffer_length ()

```
guint
ags_peak_util_get_buffer_length (AgsPeakUtil *peak_util);
```

Get buffer length of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	

Returns

the buffer length

Since: 3.9.2

ags_peak_util_set_buffer_length ()

```
void
ags_peak_util_set_buffer_length (AgsPeakUtil *peak_util,
                                 guint buffer_length);
```

Set *buffer_length* of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	
buffer_length	the buffer length	

Since: 3.9.2

ags_peak_util_get_format ()

```
guint
ags_peak_util_get_format (AgsPeakUtil *peak_util);
```

Get format of *peak_util* .

Parameters

peak_util	the AgsPeakUtil

Returns

the format

Since: 3.9.6

ags_peak_util_set_format ()

```
void
ags_peak_util_set_format (AgsPeakUtil *peak_util,
                          guint format);
```

Set *format* of *peak_util* .

Parameters

peak_util	the AgsPeakUtil
format	the format

Since: 3.9.6

ags_peak_util_get_samplerate ()

```
guint
ags_peak_util_get_samplerate (AgsPeakUtil *peak_util);
```

Get samplerate of *peak_util* .

Parameters

peak_util	the AgsPeakUtil

Returns

the samplerate

Since: 3.9.2

ags_peak_util_set_samplerate ()

```
void
ags_peak_util_set_samplerate (AgsPeakUtil *peak_util,
                              guint samplerate);
```

Set *samplerate* of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	
samplerate	the samplerate	

Since: 3.9.2

ags_peak_util_get_harmonic_rate ()

```
gdouble
ags_peak_util_get_harmonic_rate (AgsPeakUtil *peak_util);
```

Get harmonic rate of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	

Returns

the harmonic rate

Since: 3.9.2

ags_peak_util_set_harmonic_rate ()

```
void
ags_peak_util_set_harmonic_rate (AgsPeakUtil *peak_util,
                                 gdouble harmonic_rate);
```

Set *harmonic_rate* of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	
harmonic_rate	the harmonic rate	

Since: 3.9.2

ags_peak_util_get_pressure_factor ()

```
gdouble
ags_peak_util_get_pressure_factor (AgsPeakUtil *peak_util);
```

Get pressure factor of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	

Returns

the pressure factor

Since: 3.9.2

ags_peak_util_set_pressure_factor ()

```
void
ags_peak_util_set_pressure_factor (AgsPeakUtil *peak_util,
                                   gdouble pressure_factor);
```

Set *pressure_factor* of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	
pressure_factor	the pressure factor	

Since: 3.9.2

ags_peak_util_get_peak ()

```
gdouble
ags_peak_util_get_peak (AgsPeakUtil *peak_util);
```

Get peak of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	

Returns

the peak

Since: 3.9.2

ags_peak_util_set_peak ()

```
void
ags_peak_util_set_peak (AgsPeakUtil *peak_util,
                        gdouble peak);
```

Set *peak* of *peak_util* .

Parameters

peak_util	the AgsPeakUtil	
peak	the peak	

Since: 3.9.2

ags_peak_util_compute_s8 ()

```
void
ags_peak_util_compute_s8 (AgsPeakUtil *peak_util);
```

Compute peak of signed 8 bit data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_s16 ()

```
void
ags_peak_util_compute_s16 (AgsPeakUtil *peak_util);
```

Compute peak of signed 16 bit data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_s24 ()

```
void
ags_peak_util_compute_s24 (AgsPeakUtil *peak_util);
```

Compute peak of signed 24 bit data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_s32 ()

```
void
ags_peak_util_compute_s32 (AgsPeakUtil *peak_util);
```

Compute peak of signed 32 bit data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_s64 ()

```
void
ags_peak_util_compute_s64 (AgsPeakUtil *peak_util);
```

Compute peak of signed 64 bit data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_float ()

```
void
ags_peak_util_compute_float (AgsPeakUtil *peak_util);
```

Compute peak of floating point data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_double ()

```
void
ags_peak_util_compute_double (AgsPeakUtil *peak_util);
```

Compute peak of double floating point data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute_complex ()

```
void
ags_peak_util_compute_complex (AgsPeakUtil *peak_util);
```

Compute peak of complex floating point data.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

ags_peak_util_compute ()

```
void
ags_peak_util_compute (AgsPeakUtil *peak_util);
```

Compute peak.

Parameters

peak_util	the AgsPeakUtil

Since: 3.9.2

AGS_PEAK_UTIL()

```
#define AGS_PEAK_UTIL(ptr) ((AgsPeakUtil *)(ptr))
```

ags_peak_util_get_type ()

```
GType
ags_peak_util_get_type (void);
```

Types and Values

AGS_PEAK_UTIL_DEFAULT_FORMAT

```
#define AGS_PEAK_UTIL_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_PEAK_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT

```
#define AGS_PEAK_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT (AGS_AUDIO_BUFFER_UTIL_S16)
```

AGS_PEAK_UTIL_DEFAULT_SAMPLERATE

```
#define AGS_PEAK_UTIL_DEFAULT_SAMPLERATE (AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_PEAK_UTIL_DEFAULT_HARMONIC_RATE

```
#define AGS_PEAK_UTIL_DEFAULT_HARMONIC_RATE (440.0)
```

AGS_PEAK_UTIL_DEFAULT_PRESSURE_FACTOR

```
#define AGS_PEAK_UTIL_DEFAULT_PRESSURE_FACTOR (1.0)
```

AGS_TYPE_PEAK_UTIL

```
#define AGS_TYPE_PEAK_UTIL          (ags_peak_util_get_type())
```

struct AgsPeakUtil

```
struct AgsPeakUtil {
  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble harmonic_rate;
  gdouble pressure_factor;

  gdouble peak;
};
```

5.18 AgsNoiseUtil

AgsNoiseUtil — noise util

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_NOISE_UTIL_RESET()
#define	AGS_NOISE_UTIL_GET_UNSCALED_VALUE()
#define	AGS_NOISE_UTIL_GET_VALUE()
AgsNoiseUtil *	ags_noise_util_alloc ()
gpointer	ags_noise_util_copy ()
void	ags_noise_util_free ()
gpointer	ags_noise_util_get_source ()
void	ags_noise_util_set_source ()
guint	ags_noise_util_get_source_stride ()
void	ags_noise_util_set_source_stride ()
gpointer	ags_noise_util_get_destination ()
void	ags_noise_util_set_destination ()
guint	ags_noise_util_get_destination_stride ()
void	ags_noise_util_set_destination_stride ()
guint	ags_noise_util_get_buffer_length ()
void	ags_noise_util_set_buffer_length ()

guint	ags_noise_util_get_format ()
void	ags_noise_util_set_format ()
guint	ags_noise_util_get_samplerate ()
void	ags_noise_util_set_samplerate ()
gdouble	ags_noise_util_get_volume ()
void	ags_noise_util_set_volume ()
gdouble	ags_noise_util_get_frequency ()
void	ags_noise_util_set_frequency ()
guint	ags_noise_util_get_frame_count ()
void	ags_noise_util_set_frame_count ()
guint	ags_noise_util_get_offset ()
void	ags_noise_util_set_offset ()
void	ags_noise_util_compute_s8 ()
void	ags_noise_util_compute_s16 ()
void	ags_noise_util_compute_s24 ()
void	ags_noise_util_compute_s32 ()
void	ags_noise_util_compute_s64 ()
void	ags_noise_util_compute_float ()
void	ags_noise_util_compute_double ()
void	ags_noise_util_compute_complex ()
void	ags_noise_util_compute ()
GType	ags_noise_util_get_type ()

Types and Values

#define	AGS_NOISE_UTIL_DEFAULT_FREQUENCY
enum	AgsNoiseuUtilMode
#define	AGS_TYPE_NOISE_UTIL
struct	AgsNoiseUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsNoiseUtil
```

Includes

```
#include <ags/audio/ags_noise_util.h>
```

Description

Utility functions to compute noise.

Functions

AGS_NOISE_UTIL_RESET()

```
#define             AGS_NOISE_UTIL_RESET(noise, ptr_last_value)
```

AGS_NOISE_UTIL_GET_UNSCALED_VALUE()

```
#define             AGS_NOISE_UTIL_GET_UNSCALED_VALUE(unscaled_value, noise, ptr_counter, ↩
    ptr_last_value)
```

AGS_NOISE_UTIL_GET_VALUE()

```
#define             AGS_NOISE_UTIL_GET_VALUE(value, noise, ptr_counter, ptr_last_value)
```

ags_noise_util_alloc ()

```
AgsNoiseUtil~*
ags_noise_util_alloc ();
```

Allocate AgsNoiseUtil.

Returns

the newly allocated AgsNoiseUtil

Since: 3.15.0

ags_noise_util_copy ()

```
gpointer
ags_noise_util_copy (AgsNoiseUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsNoiseUtil

Returns

a pointer of the new AgsNoiseUtil

Since: 3.15.0

ags_noise_util_free ()

```
void
ags_noise_util_free (AgsNoiseUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_get_source ()

```
gpointer
ags_noise_util_get_source (AgsNoiseUtil *noise_util);
```

Get source buffer of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil

Returns

the source buffer

Since: 3.15.0

ags_noise_util_set_source ()

```
void
ags_noise_util_set_source (AgsNoiseUtil *noise_util,
                           gpointer source);
```

Set *source* buffer of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil
source	the source buffer

Since: 3.15.0

ags_noise_util_get_source_stride ()

```
guint
ags_noise_util_get_source_stride (AgsNoiseUtil *noise_util);
```

Get source stride of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil

Returns

the source buffer stride

Since: 3.15.0

ags_noise_util_set_source_stride ()

```
void
ags_noise_util_set_source_stride (AgsNoiseUtil *noise_util,
                                  guint source_stride);
```

Set *source* stride of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
source_stride	the source buffer stride	

Since: 3.15.0

ags_noise_util_get_destination ()

```
gpointer
ags_noise_util_get_destination (AgsNoiseUtil *noise_util);
```

Get destination buffer of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the destination buffer

Since: 3.15.0

ags_noise_util_set_destination ()

```
void
ags_noise_util_set_destination (AgsNoiseUtil *noise_util,
                                gpointer destination);
```

Set *destination* buffer of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
destination	the destination buffer	

Since: 3.15.0

ags_noise_util_get_destination_stride ()

```
guint
ags_noise_util_get_destination_stride (AgsNoiseUtil *noise_util);
```

Get destination stride of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the destination buffer stride

Since: 3.15.0

ags_noise_util_set_destination_stride ()

```
void
ags_noise_util_set_destination_stride (AgsNoiseUtil *noise_util,
                                       guint destination_stride);
```

Set *destination* stride of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
destination_stride	the destination buffer stride	

Since: 3.15.0

ags_noise_util_get_buffer_length ()

```
guint
ags_noise_util_get_buffer_length (AgsNoiseUtil *noise_util);
```

Get buffer length of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the buffer length

Since: 3.15.0

ags_noise_util_set_buffer_length ()

```
void
ags_noise_util_set_buffer_length (AgsNoiseUtil *noise_util,
                                  guint buffer_length);
```

Set *buffer_length* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
buffer_length	the buffer length	

Since: 3.15.0

ags_noise_util_get_format ()

```
guint
ags_noise_util_get_format (AgsNoiseUtil *noise_util);
```

Get format of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the format

Since: 3.15.0

ags_noise_util_set_format ()

```
void
ags_noise_util_set_format (AgsNoiseUtil *noise_util,
                           guint format);
```

Set *format* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
format	the format	

Since: 3.15.0

ags_noise_util_get_samplerate ()

```
guint
ags_noise_util_get_samplerate (AgsNoiseUtil *noise_util);
```

Get samplerate of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the samplerate

Since: 3.15.0

ags_noise_util_set_samplerate ()

```
void
ags_noise_util_set_samplerate (AgsNoiseUtil *noise_util,
                               guint samplerate);
```

Set *samplerate* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
samplerate	the samplerate	

Since: 3.15.0

ags_noise_util_get_volume ()

```
gdouble
ags_noise_util_get_volume (AgsNoiseUtil *noise_util);
```

Get volume of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the volume

Since: 3.15.0

ags_noise_util_set_volume ()

```
void
ags_noise_util_set_volume (AgsNoiseUtil *noise_util,
                           gdouble volume);
```

Set *volume* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
volume	the volume	

Since: 3.15.0

ags_noise_util_get_frequency ()

```
gdouble
ags_noise_util_get_frequency (AgsNoiseUtil *noise_util);
```

Get frequency of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil

Returns

the frequency

Since: 3.15.0

ags_noise_util_set_frequency ()

```
void
ags_noise_util_set_frequency (AgsNoiseUtil *noise_util,
                              gdouble frequency);
```

Set *frequency* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil
frequency	the frequency

Since: 3.15.0

ags_noise_util_get_frame_count ()

```
guint
ags_noise_util_get_frame_count (AgsNoiseUtil *noise_util);
```

Get frame count of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil

Returns

the frame count

Since: 3.15.0

ags_noise_util_set_frame_count ()

```
void
ags_noise_util_set_frame_count (AgsNoiseUtil *noise_util,
                                guint frame_count);
```

Set *frame_count* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
frame_count	the frame count	

Since: 3.15.0

ags_noise_util_get_offset ()

```
guint
ags_noise_util_get_offset (AgsNoiseUtil *noise_util);
```

Get offset of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	

Returns

the offset

Since: 3.15.0

ags_noise_util_set_offset ()

```
void
ags_noise_util_set_offset (AgsNoiseUtil *noise_util,
                           guint offset);
```

Set *offset* of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil	
offset	the offset	

Since: 3.15.0

ags_noise_util_compute_s8 ()

```
void
ags_noise_util_compute_s8 (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of signed 8 bit data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_s16 ()

```
void
ags_noise_util_compute_s16 (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of signed 16 bit data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_s24 ()

```
void
ags_noise_util_compute_s24 (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of signed 24 bit data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_s32 ()

```
void
ags_noise_util_compute_s32 (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of signed 32 bit data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_s64 ()

```
void
ags_noise_util_compute_s64 (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of signed 64 bit data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_float ()

```
void
ags_noise_util_compute_float (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of floating point data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_double ()

```
void
ags_noise_util_compute_double (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of double precision floating point data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute_complex ()

```
void
ags_noise_util_compute_complex (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* of complex data.

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_compute ()

```
void
ags_noise_util_compute (AgsNoiseUtil *noise_util);
```

Compute noise of *noise_util* .

Parameters

noise_util	the AgsNoiseUtil

Since: 3.15.0

ags_noise_util_get_type ()

```
GType
ags_noise_util_get_type (void);
```

Types and Values

AGS_NOISE_UTIL_DEFAULT_FREQUENCY

```
#define AGS_NOISE_UTIL_DEFAULT_FREQUENCY (440.0)
```

enum AgsNoiseuUtilMode

Enum values to specify noise mode.

Members

AGS_NOISE_UTIL_WHITE_NOISE	white noise
AGS_NOISE_UTIL_PINK_NOISE	pink noise

AGS_TYPE_NOISE_UTIL

```
#define AGS_TYPE_NOISE_UTIL         (ags_noise_util_get_type())
```

struct AgsNoiseUtil

```
struct AgsNoiseUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  gpointer noise;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint mode;

  gdouble volume;
```

```
  gdouble frequency;

  guint frame_count;
  guint offset;
};
```

5.19 AgsResampleUtil

AgsResampleUtil — Boxed type of resample util

Stability Level

Stable, unless otherwise indicated

Functions

AgsResampleUtil *	ags_resample_util_alloc ()
gpointer	ags_resample_util_copy ()
void	ags_resample_util_free ()
gpointer	ags_resample_util_get_destination ()
void	ags_resample_util_set_destination ()
guint	ags_resample_util_get_destination_stride ()
void	ags_resample_util_set_destination_stride ()
gpointer	ags_resample_util_get_source ()
void	ags_resample_util_set_source ()
guint	ags_resample_util_get_source_stride ()
void	ags_resample_util_set_source_stride ()
guint	ags_resample_util_get_buffer_length ()
void	ags_resample_util_set_buffer_length ()
guint	ags_resample_util_get_format ()
void	ags_resample_util_set_format ()
guint	ags_resample_util_get_samplerate ()
void	ags_resample_util_set_samplerate ()
guint	ags_resample_util_get_target_samplerate ()
void	ags_resample_util_set_target_samplerate ()
void	ags_resample_util_compute_s8 ()
void	ags_resample_util_compute_s16 ()
void	ags_resample_util_compute_s24 ()
void	ags_resample_util_compute_s32 ()
void	ags_resample_util_compute_s64 ()
void	ags_resample_util_compute_float ()
void	ags_resample_util_compute_double ()
void	ags_resample_util_compute_complex ()
void	ags_resample_util_compute ()
#define	AGS_RESAMPLE_UTIL()
GType	ags_resample_util_get_type ()

Types and Values

#define	AGS_RESAMPLE_UTIL_DEFAULT_FORMAT
#define	AGS_RESAMPLE_UTIL_DEFAULT_SAMPLERATE

#define	AGS_RESAMPLE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT
#define	AGS_RESAMPLE_UTIL_DEFAULT_TARGET_SAMPLERATE
#define	AGS_TYPE_RESAMPLE_UTIL
struct	AgsResampleUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsResampleUtil
```

Includes

```
#include <ags/audio/ags_resample_util.h>
```

Description

Boxed type of resample util data type.

Functions

ags_resample_util_alloc ()

```
AgsResampleUtil~*
ags_resample_util_alloc ();
```

Allocate AgsResampleUtil

Returns

a new AgsResampleUtil

Since: 3.9.2

ags_resample_util_copy ()

```
gpointer
ags_resample_util_copy (AgsResampleUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsResampleUtil

Returns

a pointer of the new AgsResampleUtil

Since: 3.9.2

ags_resample_util_free ()

```
void
ags_resample_util_free (AgsResampleUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_get_destination ()

```
gpointer
ags_resample_util_get_destination (AgsResampleUtil *resample_util);
```

Get destination buffer of *resample_util* .

Parameters

resample_util	the AgsResampleUtil

Returns

the destination buffer

Since: 3.9.2

ags_resample_util_set_destination ()

```
void
ags_resample_util_set_destination (AgsResampleUtil *resample_util,
                                   gpointer destination);
```

Set *destination* buffer of *resample_util* .

Parameters

resample_util	the AgsResampleUtil
destination	the destination buffer

Since: 3.9.2

ags_resample_util_get_destination_stride ()

```
guint
ags_resample_util_get_destination_stride
                                (AgsResampleUtil *resample_util);
```

Get destination stride of *resample_util* .

Parameters

resample_util	the AgsResampleUtil

Returns

the destination buffer stride

Since: 3.9.2

ags_resample_util_set_destination_stride ()

```
void
ags_resample_util_set_destination_stride
                                (AgsResampleUtil *resample_util,
                                 guint destination_stride);
```

Set *destination* stride of *resample_util* .

Parameters

resample_util	the AgsResampleUtil
destination_stride	the destination buffer stride

Since: 3.9.2

ags_resample_util_get_source ()

```
gpointer
ags_resample_util_get_source (AgsResampleUtil *resample_util);
```

Get source buffer of *resample_util* .

Parameters

resample_util	the AgsResampleUtil

Returns

the source buffer

Since: 3.9.2

ags_resample_util_set_source ()

```
void
ags_resample_util_set_source (AgsResampleUtil *resample_util,
                              gpointer source);
```

Set *source* buffer of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	
source	the source buffer	

Since: 3.9.2

ags_resample_util_get_source_stride ()

```
guint
ags_resample_util_get_source_stride (AgsResampleUtil *resample_util);
```

Get source stride of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	

Returns

the source buffer stride

Since: 3.9.2

ags_resample_util_set_source_stride ()

```
void
ags_resample_util_set_source_stride (AgsResampleUtil *resample_util,
                                     guint source_stride);
```

Set *source* stride of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	
source_stride	the source buffer stride	

Since: 3.9.2

ags_resample_util_get_buffer_length ()

```
guint
ags_resample_util_get_buffer_length (AgsResampleUtil *resample_util);
```

Get buffer length of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	

Returns

the buffer length

Since: 3.9.2

ags_resample_util_set_buffer_length ()

```
void
ags_resample_util_set_buffer_length (AgsResampleUtil *resample_util,
                                     guint buffer_length);
```

Set *buffer_length* of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	
buffer_length	the buffer length	

Since: 3.9.2

ags_resample_util_get_format ()

```
guint
ags_resample_util_get_format (AgsResampleUtil *resample_util);
```

Get format of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	

Returns

the format

Since: 3.9.6

ags_resample_util_set_format ()

```
void
ags_resample_util_set_format (AgsResampleUtil *resample_util,
                              guint format);
```

Set *format* of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	
format	the format	

Since: 3.9.6

ags_resample_util_get_samplerate ()

```
guint
ags_resample_util_get_samplerate (AgsResampleUtil *resample_util);
```

Get samplerate of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	

Returns

the samplerate

Since: 3.9.2

ags_resample_util_set_samplerate ()

```
void
ags_resample_util_set_samplerate (AgsResampleUtil *resample_util,
                                  guint samplerate);
```

Set *samplerate* of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	
samplerate	the samplerate	

Since: 3.9.2

ags_resample_util_get_target_samplerate ()

```
guint
ags_resample_util_get_target_samplerate
                                (AgsResampleUtil *resample_util);
```

Get target samplerate of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	

Returns

the samplerate

Since: 3.9.2

ags_resample_util_set_target_samplerate ()

```
void
ags_resample_util_set_target_samplerate
                                (AgsResampleUtil *resample_util,
                                 guint target_samplerate);
```

Set *target_samplerate* of *resample_util* .

Parameters

resample_util	the AgsResampleUtil	
target_samplerate	the target samplerate	

Since: 3.9.2

ags_resample_util_compute_s8 ()

```
void
ags_resample_util_compute_s8 (AgsResampleUtil *resample_util);
```

Compute resample of signed 8 bit data.

Parameters

resample_util	the AgsResampleUtil	

Since: 3.9.2

ags_resample_util_compute_s16 ()

```
void
ags_resample_util_compute_s16 (AgsResampleUtil *resample_util);
```

Compute resample of signed 16 bit data.

Parameters

resample_util	the AgsResampleUtil	

Since: 3.9.2

ags_resample_util_compute_s24 ()

```
void
ags_resample_util_compute_s24 (AgsResampleUtil *resample_util);
```

Compute resample of signed 24 bit data.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_compute_s32 ()

```
void
ags_resample_util_compute_s32 (AgsResampleUtil *resample_util);
```

Compute resample of signed 32 bit data.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_compute_s64 ()

```
void
ags_resample_util_compute_s64 (AgsResampleUtil *resample_util);
```

Compute resample of signed 64 bit data.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_compute_float ()

```
void
ags_resample_util_compute_float (AgsResampleUtil *resample_util);
```

Compute resample of floating point data.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_compute_double ()

```
void
ags_resample_util_compute_double (AgsResampleUtil *resample_util);
```

Compute resample of double floating point data.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_compute_complex ()

```
void
ags_resample_util_compute_complex (AgsResampleUtil *resample_util);
```

Compute resample of complex floating point data.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

ags_resample_util_compute ()

```
void
ags_resample_util_compute (AgsResampleUtil *resample_util);
```

Compute resample.

Parameters

resample_util	the AgsResampleUtil

Since: 3.9.2

AGS_RESAMPLE_UTIL()

```
#define AGS_RESAMPLE_UTIL(ptr) ((AgsResampleUtil *)(ptr))
```

ags_resample_util_get_type ()

```
GType
ags_resample_util_get_type (void);
```

Types and Values

AGS_RESAMPLE_UTIL_DEFAULT_FORMAT

```
#define AGS_RESAMPLE_UTIL_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_RESAMPLE_UTIL_DEFAULT_SAMPLERATE

```
#define AGS_RESAMPLE_UTIL_DEFAULT_SAMPLERATE (AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_RESAMPLE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT

```
#define AGS_RESAMPLE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT (AGS_AUDIO_BUFFER_UTIL_S16)
```

AGS_RESAMPLE_UTIL_DEFAULT_TARGET_SAMPLERATE

```
#define AGS_RESAMPLE_UTIL_DEFAULT_TARGET_SAMPLERATE (AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_TYPE_RESAMPLE_UTIL

```
#define AGS_TYPE_RESAMPLE_UTIL         (ags_resample_util_get_type())
```

struct AgsResampleUtil

```
struct AgsResampleUtil {
  SRC_DATA secret_rabbit;

  gpointer destination;
  guint destination_stride;

  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint target_samplerate;
};
```

5.20 AgsSynthUtil

AgsSynthUtil — synth util

Stability Level

Stable, unless otherwise indicated

Functions

AgsSynthUtil *	ags_synth_util_alloc ()
gpointer	ags_synth_util_copy ()
void	ags_synth_util_free ()
gpointer	ags_synth_util_get_source ()
void	ags_synth_util_set_source ()
guint	ags_synth_util_get_source_stride ()
void	ags_synth_util_set_source_stride ()
guint	ags_synth_util_get_buffer_length ()
void	ags_synth_util_set_buffer_length ()
guint	ags_synth_util_get_format ()
void	ags_synth_util_set_format ()
guint	ags_synth_util_get_samplerate ()
void	ags_synth_util_set_samplerate ()
guint	ags_synth_util_get_synth_oscillator_mode ()
void	ags_synth_util_set_synth_oscillator_mode ()
gdouble	ags_synth_util_get_frequency ()
void	ags_synth_util_set_frequency ()
gdouble	ags_synth_util_get_phase ()
void	ags_synth_util_set_phase ()
gdouble	ags_synth_util_get_volume ()
void	ags_synth_util_set_volume ()
guint	ags_synth_util_get_frame_count ()
void	ags_synth_util_set_frame_count ()
guint	ags_synth_util_get_offset ()
void	ags_synth_util_set_offset ()
guint	ags_synth_util_get_xcross_count_s8 ()
guint	ags_synth_util_get_xcross_count_s16 ()
guint	ags_synth_util_get_xcross_count_s24 ()
guint	ags_synth_util_get_xcross_count_s32 ()
guint	ags_synth_util_get_xcross_count_s64 ()
guint	ags_synth_util_get_xcross_count_float ()
guint	ags_synth_util_get_xcross_count_double ()
guint	ags_synth_util_get_xcross_count_complex ()
guint	ags_synth_util_get_xcross_count ()
void	ags_synth_util_compute_sin_s8 ()
void	ags_synth_util_compute_sin_s16 ()
void	ags_synth_util_compute_sin_s24 ()
void	ags_synth_util_compute_sin_s32 ()
void	ags_synth_util_compute_sin_s64 ()
void	ags_synth_util_compute_sin_float ()
void	ags_synth_util_compute_sin_double ()
void	ags_synth_util_compute_sin_complex ()
void	ags_synth_util_compute_sin ()
void	ags_synth_util_compute_sawtooth_s8 ()
void	ags_synth_util_compute_sawtooth_s16 ()
void	ags_synth_util_compute_sawtooth_s24 ()
void	ags_synth_util_compute_sawtooth_s32 ()
void	ags_synth_util_compute_sawtooth_s64 ()
void	ags_synth_util_compute_sawtooth_float ()
void	ags_synth_util_compute_sawtooth_double ()
void	ags_synth_util_compute_sawtooth_complex ()
void	ags_synth_util_compute_sawtooth ()
void	ags_synth_util_compute_triangle_s8 ()
void	ags_synth_util_compute_triangle_s16 ()

void	ags_synth_util_compute_triangle_s24 ()
void	ags_synth_util_compute_triangle_s32 ()
void	ags_synth_util_compute_triangle_s64 ()
void	ags_synth_util_compute_triangle_float ()
void	ags_synth_util_compute_triangle_double ()
void	ags_synth_util_compute_triangle_complex ()
void	ags_synth_util_compute_triangle ()
void	ags_synth_util_compute_square_s8 ()
void	ags_synth_util_compute_square_s16 ()
void	ags_synth_util_compute_square_s24 ()
void	ags_synth_util_compute_square_s32 ()
void	ags_synth_util_compute_square_s64 ()
void	ags_synth_util_compute_square_float ()
void	ags_synth_util_compute_square_double ()
void	ags_synth_util_compute_square_complex ()
void	ags_synth_util_compute_square ()
void	ags_synth_util_compute_impulse_s8 ()
void	ags_synth_util_compute_impulse_s16 ()
void	ags_synth_util_compute_impulse_s24 ()
void	ags_synth_util_compute_impulse_s32 ()
void	ags_synth_util_compute_impulse_s64 ()
void	ags_synth_util_compute_impulse_float ()
void	ags_synth_util_compute_impulse_double ()
void	ags_synth_util_compute_impulse_complex ()
void	ags_synth_util_compute_impulse ()
GType	ags_synth_util_get_type ()

Types and Values

#define	AGS_SYNTH_UTIL_DEFAULT_FREQUENCY
#define	AGS_TYPE_SYNTH_UTIL
struct	AgsSynthUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsSynthUtil
```

Includes

```
#include <ags/audio/ags_synth_util.h>
```

Description

Utility functions to compute synths.

Functions

ags_synth_util_alloc ()

```
AgsSynthUtil~*
ags_synth_util_alloc ();
```

Allocate AgsSynthUtil

Returns

a new AgsSynthUtil

Since: 3.9.3

ags_synth_util_copy ()

```
gpointer
ags_synth_util_copy (AgsSynthUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsSynthUtil

Returns

a pointer of the new AgsSynthUtil

Since: 3.9.3

ags_synth_util_free ()

```
void
ags_synth_util_free (AgsSynthUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_get_source ()

```
gpointer
ags_synth_util_get_source (AgsSynthUtil *synth_util);
```

Get source buffer of *synth_util* .

Parameters

synth_util	the AgsSynthUtil

Returns

the source buffer

Since: 3.9.3

ags_synth_util_set_source ()

```
void
ags_synth_util_set_source (AgsSynthUtil *synth_util,
                           gpointer source);
```

Set *source* buffer of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
source	the source buffer	

Since: 3.9.3

ags_synth_util_get_source_stride ()

```
guint
ags_synth_util_get_source_stride (AgsSynthUtil *synth_util);
```

Get source stride of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the source buffer stride

Since: 3.9.3

ags_synth_util_set_source_stride ()

```
void
ags_synth_util_set_source_stride (AgsSynthUtil *synth_util,
                                  guint source_stride);
```

Set *source* stride of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
source_stride	the source buffer stride	

Since: 3.9.3

ags_synth_util_get_buffer_length ()

```
guint
ags_synth_util_get_buffer_length (AgsSynthUtil *synth_util);
```

Get buffer length of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the buffer length

Since: 3.9.3

ags_synth_util_set_buffer_length ()

```
void
ags_synth_util_set_buffer_length (AgsSynthUtil *synth_util,
                                  guint buffer_length);
```

Set *buffer_length* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
buffer_length	the buffer length	

Since: 3.9.3

ags_synth_util_get_format ()

```
guint
ags_synth_util_get_format (AgsSynthUtil *synth_util);
```

Get audio buffer util format of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the audio buffer util format

Since: 3.14.0

ags_synth_util_set_format ()

```
void
ags_synth_util_set_format (AgsSynthUtil *synth_util,
                           guint format);
```

Set *format* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
format	the audio buffer util format	

Since: 3.14.0

ags_synth_util_get_samplerate ()

```
guint
ags_synth_util_get_samplerate (AgsSynthUtil *synth_util);
```

Get samplerate of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the samplerate

Since: 3.9.3

ags_synth_util_set_samplerate ()

```
void
ags_synth_util_set_samplerate (AgsSynthUtil *synth_util,
                               guint samplerate);
```

Set *samplerate* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
samplerate	the samplerate	

Since: 3.9.3

ags_synth_util_get_synth_oscillator_mode ()

```
guint
ags_synth_util_get_synth_oscillator_mode
                               (AgsSynthUtil *synth_util);
```

Get synth oscillator mode of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the synth oscillator mode

Since: 3.9.3

ags_synth_util_set_synth_oscillator_mode ()

```
void
ags_synth_util_set_synth_oscillator_mode
                                (AgsSynthUtil *synth_util,
                                 guint synth_oscillator_mode);
```

Set *synth_oscillator_mode* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
synth_oscillator_mode	the synth oscillator mode	

Since: 3.9.3

ags_synth_util_get_frequency ()

```
gdouble
ags_synth_util_get_frequency (AgsSynthUtil *synth_util);
```

Get frequency of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the frequency

Since: 3.9.3

ags_synth_util_set_frequency ()

```
void
ags_synth_util_set_frequency (AgsSynthUtil *synth_util,
                              gdouble frequency);
```

Set *frequency* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
frequency	the frequency	

Since: 3.9.3

ags_synth_util_get_phase ()

```
gdouble
ags_synth_util_get_phase (AgsSynthUtil *synth_util);
```

Get phase of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the phase

Since: 3.9.3

ags_synth_util_set_phase ()

```
void
ags_synth_util_set_phase (AgsSynthUtil *synth_util,
                          gdouble phase);
```

Set *phase* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
phase	the phase	

Since: 3.9.3

ags_synth_util_get_volume ()

```
gdouble
ags_synth_util_get_volume (AgsSynthUtil *synth_util);
```

Get volume of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the volume

Since: 3.9.3

ags_synth_util_set_volume ()

```
void
ags_synth_util_set_volume (AgsSynthUtil *synth_util,
                           gdouble volume);
```

Set *volume* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
volume	the volume	

Since: 3.9.3

ags_synth_util_get_frame_count ()

```
guint
ags_synth_util_get_frame_count (AgsSynthUtil *synth_util);
```

Get frame count of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the frame count

Since: 3.9.3

ags_synth_util_set_frame_count ()

```
void
ags_synth_util_set_frame_count (AgsSynthUtil *synth_util,
                                guint frame_count);
```

Set *frame_count* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
frame_count	the frame count	

Since: 3.9.3

ags_synth_util_get_offset ()

```
guint
ags_synth_util_get_offset (AgsSynthUtil *synth_util);
```

Get offset of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	

Returns

the offset

Since: 3.9.3

ags_synth_util_set_offset ()

```
void
ags_synth_util_set_offset (AgsSynthUtil *synth_util,
                           guint offset);
```

Set *offset* of *synth_util* .

Parameters

synth_util	the AgsSynthUtil	
offset	the offset	

Since: 3.9.3

ags_synth_util_get_xcross_count_s8 ()

```
guint
ags_synth_util_get_xcross_count_s8 (gint8 *buffer,
                                    guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_s16 ()

```
guint
ags_synth_util_get_xcross_count_s16 (gint16 *buffer,
                                     guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_s24 ()

```
guint
ags_synth_util_get_xcross_count_s24 (gint32 *buffer,
                                     guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_s32 ()

```
guint
ags_synth_util_get_xcross_count_s32 (gint32 *buffer,
                                     guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_s64 ()

```
guint
ags_synth_util_get_xcross_count_s64 (gint64 *buffer,
                                     guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_float ()

```
guint
ags_synth_util_get_xcross_count_float (gfloat *buffer,
                                       guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_double ()

```
guint
ags_synth_util_get_xcross_count_double
                                (gdouble *buffer,
                                 guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count_complex ()

```
guint
ags_synth_util_get_xcross_count_complex
                                (AgsComplex *buffer,
                                 guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_get_xcross_count ()

```
guint
ags_synth_util_get_xcross_count (void *buffer,
                                 guint format,
                                 guint buffer_size);
```

Get zero-cross count.

Parameters

buffer	the buffer containing audio data	
format	the format	
buffer_size	the buffer size	

Returns

the count of zero-crossings

Since: 3.0.0

ags_synth_util_compute_sin_s8 ()

```
void
ags_synth_util_compute_sin_s8 (AgsSynthUtil *synth_util);
```

Compute sine synth of signed 8 bit data.

Parameters

synth_util	the AgsSynthUtil	

Since: 3.9.3

ags_synth_util_compute_sin_s16 ()

```
void
ags_synth_util_compute_sin_s16 (AgsSynthUtil *synth_util);
```

Compute sine synth of signed 16 bit data.

Parameters

synth_util	the AgsSynthUtil	

Since: 3.9.3

ags_synth_util_compute_sin_s24 ()

```
void
ags_synth_util_compute_sin_s24 (AgsSynthUtil *synth_util);
```

Compute sine synth of signed 24 bit data.

Parameters

synth_util	the AgsSynthUtil	

Since: 3.9.3

ags_synth_util_compute_sin_s32 ()

```
void
ags_synth_util_compute_sin_s32 (AgsSynthUtil *synth_util);
```

Compute sine synth of signed 32 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sin_s64 ()

```
void
ags_synth_util_compute_sin_s64 (AgsSynthUtil *synth_util);
```

Compute sine synth of signed 64 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sin_float ()

```
void
ags_synth_util_compute_sin_float (AgsSynthUtil *synth_util);
```

Compute sine synth of floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sin_double ()

```
void
ags_synth_util_compute_sin_double (AgsSynthUtil *synth_util);
```

Compute sine synth of double precision floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sin_complex ()

```
void
ags_synth_util_compute_sin_complex (AgsSynthUtil *synth_util);
```

Compute sine synth of complex data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sin ()

```
void
ags_synth_util_compute_sin (AgsSynthUtil *synth_util);
```

Compute sine synth.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_s8 ()

```
void
ags_synth_util_compute_sawtooth_s8 (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of signed 8 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_s16 ()

```
void
ags_synth_util_compute_sawtooth_s16 (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of signed 16 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_s24 ()

```
void
ags_synth_util_compute_sawtooth_s24 (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of signed 24 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_s32 ()

```
void
ags_synth_util_compute_sawtooth_s32 (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of signed 32 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_s64 ()

```
void
ags_synth_util_compute_sawtooth_s64 (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of signed 64 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_float ()

```
void
ags_synth_util_compute_sawtooth_float (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_double ()

```
void
ags_synth_util_compute_sawtooth_double
                                (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of double precision floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth_complex ()

```
void
ags_synth_util_compute_sawtooth_complex
                                (AgsSynthUtil *synth_util);
```

Compute sawtooth synth of complex data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_sawtooth ()

```
void
ags_synth_util_compute_sawtooth (AgsSynthUtil *synth_util);
```

Compute sawtooth synth.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_s8 ()

```
void
ags_synth_util_compute_triangle_s8 (AgsSynthUtil *synth_util);
```

Compute triangle synth of signed 8 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_s16 ()

```
void
ags_synth_util_compute_triangle_s16 (AgsSynthUtil *synth_util);
```

Compute triangle synth of signed 16 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_s24 ()

```
void
ags_synth_util_compute_triangle_s24 (AgsSynthUtil *synth_util);
```

Compute triangle synth of signed 24 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_s32 ()

```
void
ags_synth_util_compute_triangle_s32 (AgsSynthUtil *synth_util);
```

Compute triangle synth of signed 32 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_s64 ()

```
void
ags_synth_util_compute_triangle_s64 (AgsSynthUtil *synth_util);
```

Compute triangle synth of signed 64 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_float ()

```
void
ags_synth_util_compute_triangle_float (AgsSynthUtil *synth_util);
```

Compute triangle synth of floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_double ()

```
void
ags_synth_util_compute_triangle_double
                                (AgsSynthUtil *synth_util);
```

Compute triangle synth of double precision floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle_complex ()

```
void
ags_synth_util_compute_triangle_complex
                                (AgsSynthUtil *synth_util);
```

Compute triangle synth of complex data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_triangle ()

```
void
ags_synth_util_compute_triangle (AgsSynthUtil *synth_util);
```

Compute triangle synth.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_s8 ()

```
void
ags_synth_util_compute_square_s8 (AgsSynthUtil *synth_util);
```

Compute square synth of signed 8 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_s16 ()

```
void
ags_synth_util_compute_square_s16 (AgsSynthUtil *synth_util);
```

Compute square synth of signed 16 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_s24 ()

```
void
ags_synth_util_compute_square_s24 (AgsSynthUtil *synth_util);
```

Compute square synth of signed 24 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_s32 ()

```
void
ags_synth_util_compute_square_s32 (AgsSynthUtil *synth_util);
```

Compute square synth of signed 32 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_s64 ()

```
void
ags_synth_util_compute_square_s64 (AgsSynthUtil *synth_util);
```

Compute square synth of signed 64 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_float ()

```
void
ags_synth_util_compute_square_float (AgsSynthUtil *synth_util);
```

Compute square synth of floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_double ()

```
void
ags_synth_util_compute_square_double (AgsSynthUtil *synth_util);
```

Compute square synth of double precision floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square_complex ()

```
void
ags_synth_util_compute_square_complex (AgsSynthUtil *synth_util);
```

Compute square synth of complex data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_square ()

```
void
ags_synth_util_compute_square (AgsSynthUtil *synth_util);
```

Compute square synth.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_s8 ()

```
void
ags_synth_util_compute_impulse_s8 (AgsSynthUtil *synth_util);
```

Compute impulse synth of signed 8 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_s16 ()

```
void
ags_synth_util_compute_impulse_s16 (AgsSynthUtil *synth_util);
```

Compute impulse synth of signed 16 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_s24 ()

```
void
ags_synth_util_compute_impulse_s24 (AgsSynthUtil *synth_util);
```

Compute impulse synth of signed 24 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_s32 ()

```
void
ags_synth_util_compute_impulse_s32 (AgsSynthUtil *synth_util);
```

Compute impulse synth of signed 32 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_s64 ()

```
void
ags_synth_util_compute_impulse_s64 (AgsSynthUtil *synth_util);
```

Compute impulse synth of signed 64 bit data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_float ()

```
void
ags_synth_util_compute_impulse_float (AgsSynthUtil *synth_util);
```

Compute impulse synth of floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_double ()

```
void
ags_synth_util_compute_impulse_double (AgsSynthUtil *synth_util);
```

Compute impulse synth of double precision floating point data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse_complex ()

```
void
ags_synth_util_compute_impulse_complex
                                (AgsSynthUtil *synth_util);
```

Compute impulse synth of complex data.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_compute_impulse ()

```
void
ags_synth_util_compute_impulse (AgsSynthUtil *synth_util);
```

Compute impulse synth.

Parameters

synth_util	the AgsSynthUtil

Since: 3.9.3

ags_synth_util_get_type ()

```
GType
ags_synth_util_get_type (void);
```

Types and Values

AGS_SYNTH_UTIL_DEFAULT_FREQUENCY

```
#define AGS_SYNTH_UTIL_DEFAULT_FREQUENCY (440.0)
```

AGS_TYPE_SYNTH_UTIL

```
#define AGS_TYPE_SYNTH_UTIL         (ags_synth_util_get_type())
```

struct AgsSynthUtil

```
struct AgsSynthUtil {
  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint synth_oscillator_mode;

  gdouble frequency;
  gdouble phase;
  gdouble volume;

  guint frame_count;
  guint offset;
};
```

5.21 AgsFMSynthUtil

AgsFMSynthUtil — frequency modulation synth util

Stability Level

Stable, unless otherwise indicated

Functions

AgsFMSynthUtil *	ags_fm_synth_util_alloc ()
gpointer	ags_fm_synth_util_copy ()
void	ags_fm_synth_util_free ()
gpointer	ags_fm_synth_util_get_source ()
void	ags_fm_synth_util_set_source ()
guint	ags_fm_synth_util_get_source_stride ()
void	ags_fm_synth_util_set_source_stride ()
guint	ags_fm_synth_util_get_buffer_length ()
void	ags_fm_synth_util_set_buffer_length ()
guint	ags_fm_synth_util_get_format ()

void	ags_fm_synth_util_set_format ()
guint	ags_fm_synth_util_get_samplerate ()
void	ags_fm_synth_util_set_samplerate ()
guint	ags_fm_synth_util_get_synth_oscillator_mode ()
void	ags_fm_synth_util_set_synth_oscillator_mode ()
gdouble	ags_fm_synth_util_get_frequency ()
void	ags_fm_synth_util_set_frequency ()
gdouble	ags_fm_synth_util_get_phase ()
void	ags_fm_synth_util_set_phase ()
gdouble	ags_fm_synth_util_get_volume ()
void	ags_fm_synth_util_set_volume ()
guint	ags_fm_synth_util_get_lfo_oscillator_mode ()
void	ags_fm_synth_util_set_lfo_oscillator_mode ()
gdouble	ags_fm_synth_util_get_lfo_frequency ()
void	ags_fm_synth_util_set_lfo_frequency ()
gdouble	ags_fm_synth_util_get_lfo_depth ()
void	ags_fm_synth_util_set_lfo_depth ()
gdouble	ags_fm_synth_util_get_tuning ()
void	ags_fm_synth_util_set_tuning ()
guint	ags_fm_synth_util_get_frame_count ()
void	ags_fm_synth_util_set_frame_count ()
guint	ags_fm_synth_util_get_offset ()
void	ags_fm_synth_util_set_offset ()
void	ags_fm_synth_util_compute_sin_s8 ()
void	ags_fm_synth_util_compute_sin_s16 ()
void	ags_fm_synth_util_compute_sin_s24 ()
void	ags_fm_synth_util_compute_sin_s32 ()
void	ags_fm_synth_util_compute_sin_s64 ()
void	ags_fm_synth_util_compute_sin_float ()
void	ags_fm_synth_util_compute_sin_double ()
void	ags_fm_synth_util_compute_sin_complex ()
void	ags_fm_synth_util_compute_sin ()
void	ags_fm_synth_util_compute_sawtooth_s8 ()
void	ags_fm_synth_util_compute_sawtooth_s16 ()
void	ags_fm_synth_util_compute_sawtooth_s24 ()
void	ags_fm_synth_util_compute_sawtooth_s32 ()
void	ags_fm_synth_util_compute_sawtooth_s64 ()
void	ags_fm_synth_util_compute_sawtooth_float ()
void	ags_fm_synth_util_compute_sawtooth_double ()
void	ags_fm_synth_util_compute_sawtooth_complex ()
void	ags_fm_synth_util_compute_sawtooth ()
void	ags_fm_synth_util_compute_triangle_s8 ()
void	ags_fm_synth_util_compute_triangle_s16 ()
void	ags_fm_synth_util_compute_triangle_s24 ()
void	ags_fm_synth_util_compute_triangle_s32 ()
void	ags_fm_synth_util_compute_triangle_s64 ()
void	ags_fm_synth_util_compute_triangle_float ()
void	ags_fm_synth_util_compute_triangle_double ()
void	ags_fm_synth_util_compute_triangle_complex ()
void	ags_fm_synth_util_compute_triangle ()
void	ags_fm_synth_util_compute_square_s8 ()
void	ags_fm_synth_util_compute_square_s16 ()
void	ags_fm_synth_util_compute_square_s24 ()
void	ags_fm_synth_util_compute_square_s32 ()
void	ags_fm_synth_util_compute_square_s64 ()
void	ags_fm_synth_util_compute_square_float ()
void	ags_fm_synth_util_compute_square_double ()

void	ags_fm_synth_util_compute_square_complex ()
void	ags_fm_synth_util_compute_square ()
void	ags_fm_synth_util_compute_impulse_s8 ()
void	ags_fm_synth_util_compute_impulse_s16 ()
void	ags_fm_synth_util_compute_impulse_s24 ()
void	ags_fm_synth_util_compute_impulse_s32 ()
void	ags_fm_synth_util_compute_impulse_s64 ()
void	ags_fm_synth_util_compute_impulse_float ()
void	ags_fm_synth_util_compute_impulse_double ()
void	ags_fm_synth_util_compute_impulse_complex ()
void	ags_fm_synth_util_compute_impulse ()
GType	ags_fm_synth_util_get_type ()

Types and Values

#define	AGS_FM_SYNTH_UTIL_DEFAULT_FREQUENCY
#define	AGS_FM_SYNTH_UTIL_DEFAULT_LFO_FREQUENCY
#define	AGS_FM_SYNTH_UTIL_DEFAULT_LFO_DEPTH
#define	AGS_FM_SYNTH_UTIL_DEFAULT_TUNING
#define	AGS_TYPE_FM_SYNTH_UTIL
struct	AgsFMSynthUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFMSynthUtil
```

Includes

```
#include <ags/audio/ags_fm_synth_util.h>
```

Description

Utility functions to compute FM synths.

Functions

ags_fm_synth_util_alloc ()

```
AgsFMSynthUtil~*
ags_fm_synth_util_alloc ();
```

Allocate AgsFMSynthUtil

Returns

a new AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_copy ()

```
gpointer
ags_fm_synth_util_copy (AgsFMSynthUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFMSynthUtil

Returns

a pointer of the new AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_free ()

```
void
ags_fm_synth_util_free (AgsFMSynthUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_get_source ()

```
gpointer
ags_fm_synth_util_get_source (AgsFMSynthUtil *fm_synth_util);
```

Get source buffer of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil

Returns

the source buffer

Since: 3.9.3

ags_fm_synth_util_set_source ()

```
void
ags_fm_synth_util_set_source (AgsFMSynthUtil *fm_synth_util,
                              gpointer source);
```

Set *source* buffer of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
source	the source buffer	

Since: 3.9.3

ags_fm_synth_util_get_source_stride ()

```
guint
ags_fm_synth_util_get_source_stride (AgsFMSynthUtil *fm_synth_util);
```

Get source stride of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the source buffer stride

Since: 3.9.3

ags_fm_synth_util_set_source_stride ()

```
void
ags_fm_synth_util_set_source_stride (AgsFMSynthUtil *fm_synth_util,
                                     guint source_stride);
```

Set *source* stride of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
source_stride	the source buffer stride	

Since: 3.9.3

ags_fm_synth_util_get_buffer_length ()

```
guint
ags_fm_synth_util_get_buffer_length (AgsFMSynthUtil *fm_synth_util);
```

Get buffer length of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil

Returns

the buffer length

Since: 3.9.3

ags_fm_synth_util_set_buffer_length ()

```
void
ags_fm_synth_util_set_buffer_length (AgsFMSynthUtil *fm_synth_util,
                                     guint buffer_length);
```

Set *buffer_length* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil
buffer_length	the buffer length

Since: 3.9.3

ags_fm_synth_util_get_format ()

```
guint
ags_fm_synth_util_get_format (AgsFMSynthUtil *fm_synth_util);
```

Get audio buffer util format of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil

Returns

the audio buffer util format

Since: 3.14.0

ags_fm_synth_util_set_format ()

```
void
ags_fm_synth_util_set_format (AgsFMSynthUtil *fm_synth_util,
                              guint format);
```

Set *format* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
format	the audio buffer util format	

Since: 3.14.0

ags_fm_synth_util_get_samplerate ()

```
guint
ags_fm_synth_util_get_samplerate (AgsFMSynthUtil *fm_synth_util);
```

Get samplerate of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the samplerate

Since: 3.9.3

ags_fm_synth_util_set_samplerate ()

```
void
ags_fm_synth_util_set_samplerate (AgsFMSynthUtil *fm_synth_util,
                                  guint samplerate);
```

Set *samplerate* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
samplerate	the samplerate	

Since: 3.9.3

ags_fm_synth_util_get_synth_oscillator_mode ()

```
guint
ags_fm_synth_util_get_synth_oscillator_mode
                                (AgsFMSynthUtil *fm_synth_util);
```

Get synth oscillator mode of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the synth oscillator mode

Since: 3.9.3

ags_fm_synth_util_set_synth_oscillator_mode ()

```
void
ags_fm_synth_util_set_synth_oscillator_mode
                                (AgsFMSynthUtil *fm_synth_util,
                                 guint synth_oscillator_mode);
```

Set *synth_oscillator_mode* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
synth_oscillator_mode	the synth oscillator mode	

Since: 3.9.3

ags_fm_synth_util_get_frequency ()

```
gdouble
ags_fm_synth_util_get_frequency (AgsFMSynthUtil *fm_synth_util);
```

Get frequency of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the frequency

Since: 3.9.3

ags_fm_synth_util_set_frequency ()

```
void
ags_fm_synth_util_set_frequency (AgsFMSynthUtil *fm_synth_util,
                                 gdouble frequency);
```

Set *frequency* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
frequency	the frequency	

Since: 3.9.3

ags_fm_synth_util_get_phase ()

```
gdouble
ags_fm_synth_util_get_phase (AgsFMSynthUtil *fm_synth_util);
```

Get phase of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the phase

Since: 3.9.3

ags_fm_synth_util_set_phase ()

```
void
ags_fm_synth_util_set_phase (AgsFMSynthUtil *fm_synth_util,
                             gdouble phase);
```

Set *phase* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
phase	the phase	

Since: 3.9.3

ags_fm_synth_util_get_volume ()

```
gdouble
ags_fm_synth_util_get_volume (AgsFMSynthUtil *fm_synth_util);
```

Get volume of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the volume

Since: 3.9.3

ags_fm_synth_util_set_volume ()

```
void
ags_fm_synth_util_set_volume (AgsFMSynthUtil *fm_synth_util,
                              gdouble volume);
```

Set *volume* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
volume	the volume	

Since: 3.9.3

ags_fm_synth_util_get_lfo_oscillator_mode ()

```
guint
ags_fm_synth_util_get_lfo_oscillator_mode
                                (AgsFMSynthUtil *fm_synth_util);
```

Get LFO oscillator mode of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the LFO oscillator mode

Since: 3.9.3

ags_fm_synth_util_set_lfo_oscillator_mode ()

```
void
ags_fm_synth_util_set_lfo_oscillator_mode
                                (AgsFMSynthUtil *fm_synth_util,
                                 guint synth_oscillator_mode);
```

Set *lfo_oscillator_mode* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
lfo_oscillator_mode	the LFO oscillator mode	

Since: 3.9.3

ags_fm_synth_util_get_lfo_frequency ()

```
gdouble
ags_fm_synth_util_get_lfo_frequency (AgsFMSynthUtil *fm_synth_util);
```

Get LFO frequency of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the LFO frequency

Since: 3.9.3

ags_fm_synth_util_set_lfo_frequency ()

```
void
ags_fm_synth_util_set_lfo_frequency (AgsFMSynthUtil *fm_synth_util,
                                     gdouble lfo_frequency);
```

Set *lfo_frequency* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
lfo_frequency	the LFO frequency	

Since: 3.9.3

ags_fm_synth_util_get_lfo_depth ()

```
gdouble
ags_fm_synth_util_get_lfo_depth (AgsFMSynthUtil *fm_synth_util);
```

Get LFO depth of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the LFO depth

Since: 3.9.3

ags_fm_synth_util_set_lfo_depth ()

```
void
ags_fm_synth_util_set_lfo_depth (AgsFMSynthUtil *fm_synth_util,
                                 gdouble lfo_depth);
```

Set *lfo_depth* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
lfo_depth	the LFO depth	

Since: 3.9.3

ags_fm_synth_util_get_tuning ()

```
gdouble
ags_fm_synth_util_get_tuning (AgsFMSynthUtil *fm_synth_util);
```

Get tuning of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	

Returns

the tuning

Since: 3.9.3

ags_fm_synth_util_set_tuning ()

```
void
ags_fm_synth_util_set_tuning (AgsFMSynthUtil *fm_synth_util,
                              gdouble tuning);
```

Set *tuning* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
tuning	the tuning	

Since: 3.9.3

ags_fm_synth_util_get_frame_count ()

```
guint
ags_fm_synth_util_get_frame_count (AgsFMSynthUtil *fm_synth_util);
```

Get frame count of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil

Returns

the frame count

Since: 3.9.3

ags_fm_synth_util_set_frame_count ()

```
void
ags_fm_synth_util_set_frame_count (AgsFMSynthUtil *fm_synth_util,
                                   guint frame_count);
```

Set *frame_count* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil
frame_count	the frame count

Since: 3.9.3

ags_fm_synth_util_get_offset ()

```
guint
ags_fm_synth_util_get_offset (AgsFMSynthUtil *fm_synth_util);
```

Get offset of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil

Returns

the offset

Since: 3.9.3

ags_fm_synth_util_set_offset ()

```
void
ags_fm_synth_util_set_offset (AgsFMSynthUtil *fm_synth_util,
                              guint offset);
```

Set *offset* of *fm_synth_util* .

Parameters

fm_synth_util	the AgsFMSynthUtil	
offset	the offset	

Since: 3.9.3

ags_fm_synth_util_compute_sin_s8 ()

```
void
ags_fm_synth_util_compute_sin_s8 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of signed 8 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil	

Since: 3.9.3

ags_fm_synth_util_compute_sin_s16 ()

```
void
ags_fm_synth_util_compute_sin_s16 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of signed 16 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil	

Since: 3.9.3

ags_fm_synth_util_compute_sin_s24 ()

```
void
ags_fm_synth_util_compute_sin_s24 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of signed 24 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sin_s32 ()

```
void
ags_fm_synth_util_compute_sin_s32 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of signed 32 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sin_s64 ()

```
void
ags_fm_synth_util_compute_sin_s64 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of signed 64 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sin_float ()

```
void
ags_fm_synth_util_compute_sin_float (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sin_double ()

```
void
ags_fm_synth_util_compute_sin_double (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of double precision floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sin_complex ()

```
void
ags_fm_synth_util_compute_sin_complex (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth of complex data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sin ()

```
void
ags_fm_synth_util_compute_sin (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sine synth.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_s8 ()

```
void
ags_fm_synth_util_compute_sawtooth_s8 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of signed 8 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_s16 ()

```
void
ags_fm_synth_util_compute_sawtooth_s16
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of signed 16 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_s24 ()

```
void
ags_fm_synth_util_compute_sawtooth_s24
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of signed 24 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_s32 ()

```
void
ags_fm_synth_util_compute_sawtooth_s32
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of signed 32 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_s64 ()

```
void
ags_fm_synth_util_compute_sawtooth_s64
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of signed 64 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_float ()

```
void
ags_fm_synth_util_compute_sawtooth_float
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_double ()

```
void
ags_fm_synth_util_compute_sawtooth_double
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of double precision floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth_complex ()

```
void
ags_fm_synth_util_compute_sawtooth_complex
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth of complex data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_sawtooth ()

```
void
ags_fm_synth_util_compute_sawtooth (AgsFMSynthUtil *fm_synth_util);
```

Compute FM sawtooth synth.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_s8 ()

```
void
ags_fm_synth_util_compute_triangle_s8 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of signed 8 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_s16 ()

```
void
ags_fm_synth_util_compute_triangle_s16
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of signed 16 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_s24 ()

```
void
ags_fm_synth_util_compute_triangle_s24
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of signed 24 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_s32 ()

```
void
ags_fm_synth_util_compute_triangle_s32
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of signed 32 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_s64 ()

```
void
ags_fm_synth_util_compute_triangle_s64
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of signed 64 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_float ()

```
void
ags_fm_synth_util_compute_triangle_float
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_double ()

```
void
ags_fm_synth_util_compute_triangle_double
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of double precision floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle_complex ()

```
void
ags_fm_synth_util_compute_triangle_complex
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth of complex data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_triangle ()

```
void
ags_fm_synth_util_compute_triangle (AgsFMSynthUtil *fm_synth_util);
```

Compute FM triangle synth.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_s8 ()

```
void
ags_fm_synth_util_compute_square_s8 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of signed 8 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_s16 ()

```
void
ags_fm_synth_util_compute_square_s16 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of signed 16 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_s24 ()

```
void
ags_fm_synth_util_compute_square_s24 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of signed 24 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_s32 ()

```
void
ags_fm_synth_util_compute_square_s32 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of signed 32 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_s64 ()

```
void
ags_fm_synth_util_compute_square_s64 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of signed 64 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_float ()

```
void
ags_fm_synth_util_compute_square_float
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_double ()

```
void
ags_fm_synth_util_compute_square_double
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of double precision floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square_complex ()

```
void
ags_fm_synth_util_compute_square_complex
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth of complex data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_square ()

```
void
ags_fm_synth_util_compute_square (AgsFMSynthUtil *fm_synth_util);
```

Compute FM square synth.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_s8 ()

```
void
ags_fm_synth_util_compute_impulse_s8 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of signed 8 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_s16 ()

```
void
ags_fm_synth_util_compute_impulse_s16 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of signed 16 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_s24 ()

```
void
ags_fm_synth_util_compute_impulse_s24 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of signed 24 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_s32 ()

```
void
ags_fm_synth_util_compute_impulse_s32 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of signed 32 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_s64 ()

```
void
ags_fm_synth_util_compute_impulse_s64 (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of signed 64 bit data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_float ()

```
void
ags_fm_synth_util_compute_impulse_float
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_double ()

```
void
ags_fm_synth_util_compute_impulse_double
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of double precision floating point data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse_complex ()

```
void
ags_fm_synth_util_compute_impulse_complex
                                (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth of complex data.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_compute_impulse ()

```
void
ags_fm_synth_util_compute_impulse (AgsFMSynthUtil *fm_synth_util);
```

Compute FM impulse synth.

Parameters

fm_synth_util	the AgsFMSynthUtil

Since: 3.9.3

ags_fm_synth_util_get_type ()

```
GType
ags_fm_synth_util_get_type (void);
```

Types and Values

AGS_FM_SYNTH_UTIL_DEFAULT_FREQUENCY

```
#define AGS_FM_SYNTH_UTIL_DEFAULT_FREQUENCY (440.0)
```

AGS_FM_SYNTH_UTIL_DEFAULT_LFO_FREQUENCY

```
#define AGS_FM_SYNTH_UTIL_DEFAULT_LFO_FREQUENCY (12.0)
```

AGS_FM_SYNTH_UTIL_DEFAULT_LFO_DEPTH

```
#define AGS_FM_SYNTH_UTIL_DEFAULT_LFO_DEPTH (0.0)
```

AGS_FM_SYNTH_UTIL_DEFAULT_TUNING

```
#define AGS_FM_SYNTH_UTIL_DEFAULT_TUNING (0.0)
```

AGS_TYPE_FM_SYNTH_UTIL

```
#define AGS_TYPE_FM_SYNTH_UTIL          (ags_fm_synth_util_get_type())
```

struct AgsFMSynthUtil

```
struct AgsFMSynthUtil {
  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint synth_oscillator_mode;

  gdouble frequency;
  gdouble phase;
  gdouble volume;

  guint lfo_oscillator_mode;

  gdouble lfo_frequency;
  gdouble lfo_depth;
  gdouble tuning;

  guint frame_count;
  guint offset;
};
```

5.22 AgsSF2SynthUtil

AgsSF2SynthUtil — SF2 synth util

Stability Level

Stable, unless otherwise indicated

Functions

AgsSF2SynthUtil *	ags_sf2_synth_util_alloc ()
gpointer	ags_sf2_synth_util_boxed_copy ()
void	ags_sf2_synth_util_free ()
AgsAudioContainer *	ags_sf2_synth_util_get_sf2_file ()
void	ags_sf2_synth_util_set_sf2_file ()
gpointer	ags_sf2_synth_util_get_source ()
void	ags_sf2_synth_util_set_source ()

guint	ags_sf2_synth_util_get_source_stride ()
void	ags_sf2_synth_util_set_source_stride ()
guint	ags_sf2_synth_util_get_buffer_length ()
void	ags_sf2_synth_util_set_buffer_length ()
guint	ags_sf2_synth_util_get_format ()
void	ags_sf2_synth_util_set_format ()
guint	ags_sf2_synth_util_get_samplerate ()
void	ags_sf2_synth_util_set_samplerate ()
gchar *	ags_sf2_synth_util_get_preset ()
void	ags_sf2_synth_util_set_preset ()
gchar *	ags_sf2_synth_util_get_instrument ()
void	ags_sf2_synth_util_set_instrument ()
gchar *	ags_sf2_synth_util_get_sample ()
void	ags_sf2_synth_util_set_sample ()
gint	ags_sf2_synth_util_get_bank ()
void	ags_sf2_synth_util_set_bank ()
gint	ags_sf2_synth_util_get_program ()
void	ags_sf2_synth_util_set_program ()
gint	ags_sf2_synth_util_get_midi_key ()
void	ags_sf2_synth_util_set_midi_key ()
gdouble	ags_sf2_synth_util_get_note ()
void	ags_sf2_synth_util_set_note ()
gdouble	ags_sf2_synth_util_get_volume ()
void	ags_sf2_synth_util_set_volume ()
guint	ags_sf2_synth_util_get_frame_count ()
void	ags_sf2_synth_util_set_frame_count ()
guint	ags_sf2_synth_util_get_offset ()
void	ags_sf2_synth_util_set_offset ()
guint	ags_sf2_synth_util_get_loop_mode ()
void	ags_sf2_synth_util_set_loop_mode ()
guint	ags_sf2_synth_util_get_loop_start ()
void	ags_sf2_synth_util_set_loop_start ()
guint	ags_sf2_synth_util_get_loop_end ()
void	ags_sf2_synth_util_set_loop_end ()
AgsResampleUtil *	ags_sf2_synth_util_get_resample_util ()
void	ags_sf2_synth_util_set_resample_util ()
GType	ags_sf2_synth_util_get_pitch_type ()
void	ags_sf2_synth_util_set_pitch_type ()
gpointer	ags_sf2_synth_util_get_pitch_util ()
void	ags_sf2_synth_util_set_pitch_util ()
void	ags_sf2_synth_util_load_instrument ()
void	ags_sf2_synth_util_load_midi_locale ()
void	ags_sf2_synth_util_compute_s8 ()
void	ags_sf2_synth_util_compute_s16 ()
void	ags_sf2_synth_util_compute_s24 ()
void	ags_sf2_synth_util_compute_s32 ()
void	ags_sf2_synth_util_compute_s64 ()
void	ags_sf2_synth_util_compute_float ()
void	ags_sf2_synth_util_compute_double ()
void	ags_sf2_synth_util_compute_complex ()
void	ags_sf2_synth_util_compute ()
AgsIpatchSample *	ags_sf2_synth_util_midi_locale_find_sample_near_midi_key ()
#define	AGS_SF2_SYNTH_UTIL()
GType	ags_sf2_synth_util_get_type ()
GType	ags_sf2_synth_util_loop_mode_get_type ()

Types and Values

enum	AgsSF2SynthUtilFlags
enum	AgsSF2SynthUtilLoopMode
#define	AGS_TYPE_SF2_SYNTH_UTIL
struct	AgsSF2SynthUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsSF2SynthUtil
    GEnum
    &#x2570;&#x2500;&#x2500; AgsSF2SynthUtilLoopMode
```

Includes

```
#include <ags/audio/ags_sf2_synth_util.h>
```

Description

Utility functions to compute SF2 synths.

Functions

ags_sf2_synth_util_alloc ()

```
AgsSF2SynthUtil~*
ags_sf2_synth_util_alloc ();
```

Allocate AgsSF2SynthUtil.

Returns

the newly allocated AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_boxed_copy ()

```
gpointer
ags_sf2_synth_util_boxed_copy (AgsSF2SynthUtil *ptr);
```

Copy AgsSF2SynthUtil.

Parameters

ptr	the AgsSF2SynthUtil

Returns

the newly allocated AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_free ()

```
void
ags_sf2_synth_util_free (AgsSF2SynthUtil *ptr);
```

Free AgsSF2SynthUtil.

Parameters

ptr	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_get_sf2_file ()

```
AgsAudioContainer~*
ags_sf2_synth_util_get_sf2_file (AgsSF2SynthUtil *sf2_synth_util);
```

Get Soundfont2 file of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Returns

the Soundfont2 file.

[transfer full]

Since: 4.0.0

ags_sf2_synth_util_set_sf2_file ()

```
void
ags_sf2_synth_util_set_sf2_file (AgsSF2SynthUtil *sf2_synth_util,
                                 AgsAudioContainer *sf2_file);
```

Set Soundfont2 file of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil
sf2_file	the Soundfont2 file

Since: 4.0.0

ags_sf2_synth_util_get_source ()

```
gpointer
ags_sf2_synth_util_get_source (AgsSF2SynthUtil *sf2_synth_util);
```

Get source buffer of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Returns

the source buffer

Since: 3.9.6

ags_sf2_synth_util_set_source ()

```
void
ags_sf2_synth_util_set_source (AgsSF2SynthUtil *sf2_synth_util,
                               gpointer source);
```

Set *source* buffer of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil
source	the source buffer

Since: 3.9.6

ags_sf2_synth_util_get_source_stride ()

```
guint
ags_sf2_synth_util_get_source_stride (AgsSF2SynthUtil *sf2_synth_util);
```

Get source stride of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Returns

the source buffer stride

Since: 3.9.6

ags_sf2_synth_util_set_source_stride ()

```
void
ags_sf2_synth_util_set_source_stride (AgsSF2SynthUtil *sf2_synth_util,
                                      guint source_stride);
```

Set *source* stride of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_sf2_synth_util_get_buffer_length ()

```
guint
ags_sf2_synth_util_get_buffer_length (AgsSF2SynthUtil *sf2_synth_util);
```

Get buffer length of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the buffer length

Since: 3.9.6

ags_sf2_synth_util_set_buffer_length ()

```
void
ags_sf2_synth_util_set_buffer_length (AgsSF2SynthUtil *sf2_synth_util,
                                      guint buffer_length);
```

Set *buffer_length* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_sf2_synth_util_get_format ()

```
guint
ags_sf2_synth_util_get_format (AgsSF2SynthUtil *sf2_synth_util);
```

Get format of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the format

Since: 3.9.6

ags_sf2_synth_util_set_format ()

```
void
ags_sf2_synth_util_set_format (AgsSF2SynthUtil *sf2_synth_util,
                               guint format);
```

Set *format* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
format	the format	

Since: 3.9.6

ags_sf2_synth_util_get_samplerate ()

```
guint
ags_sf2_synth_util_get_samplerate (AgsSF2SynthUtil *sf2_synth_util);
```

Get samplerate of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the samplerate

Since: 3.9.6

ags_sf2_synth_util_set_samplerate ()

```
void
ags_sf2_synth_util_set_samplerate (AgsSF2SynthUtil *sf2_synth_util,
                                   guint samplerate);
```

Set *samplerate* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_sf2_synth_util_get_preset ()

```
gchar~*
ags_sf2_synth_util_get_preset (AgsSF2SynthUtil *sf2_synth_util);
```

Get preset of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the preset

Since: 3.9.6

ags_sf2_synth_util_set_preset ()

```
void
ags_sf2_synth_util_set_preset (AgsSF2SynthUtil *sf2_synth_util,
                               gchar *preset);
```

Set *preset* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
preset	the preset	

Since: 3.9.6

ags_sf2_synth_util_get_instrument ()

```
gchar~*
ags_sf2_synth_util_get_instrument (AgsSF2SynthUtil *sf2_synth_util);
```

Get instrument of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the instrument

Since: 3.9.6

ags_sf2_synth_util_set_instrument ()

```
void
ags_sf2_synth_util_set_instrument (AgsSF2SynthUtil *sf2_synth_util,
                                   gchar *instrument);
```

Set *instrument* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
instrument	the instrument	

Since: 3.9.6

ags_sf2_synth_util_get_sample ()

```
gchar~*
ags_sf2_synth_util_get_sample (AgsSF2SynthUtil *sf2_synth_util);
```

Get sample of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the sample

Since: 3.9.6

ags_sf2_synth_util_set_sample ()

```
void
ags_sf2_synth_util_set_sample (AgsSF2SynthUtil *sf2_synth_util,
                               gchar *sample);
```

Set *sample* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
sample	the sample	

Since: 3.9.6

ags_sf2_synth_util_get_bank ()

```
gint
ags_sf2_synth_util_get_bank (AgsSF2SynthUtil *sf2_synth_util);
```

Get bank of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the bank

Since: 3.9.6

ags_sf2_synth_util_set_bank ()

```
void
ags_sf2_synth_util_set_bank (AgsSF2SynthUtil *sf2_synth_util,
                             gint bank);
```

Set *bank* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
bank	the bank	

Since: 3.9.6

ags_sf2_synth_util_get_program ()

```
gint
ags_sf2_synth_util_get_program (AgsSF2SynthUtil *sf2_synth_util);
```

Get program of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the program

Since: 3.9.6

ags_sf2_synth_util_set_program ()

```
void
ags_sf2_synth_util_set_program (AgsSF2SynthUtil *sf2_synth_util,
                                gint program);
```

Set *program* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
program	the program	

Since: 3.9.6

ags_sf2_synth_util_get_midi_key ()

```
gint
ags_sf2_synth_util_get_midi_key (AgsSF2SynthUtil *sf2_synth_util);
```

Get midi_key of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the midi key

Since: 3.9.6

ags_sf2_synth_util_set_midi_key ()

```
void
ags_sf2_synth_util_set_midi_key (AgsSF2SynthUtil *sf2_synth_util,
                                 gint midi_key);
```

Set *midi_key* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
midi_key	the midi key	

Since: 3.9.6

ags_sf2_synth_util_get_note ()

```
gdouble
ags_sf2_synth_util_get_note (AgsSF2SynthUtil *sf2_synth_util);
```

Get note of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Returns

the note

Since: 3.9.6

ags_sf2_synth_util_set_note ()

```
void
ags_sf2_synth_util_set_note (AgsSF2SynthUtil *sf2_synth_util,
                             gdouble note);
```

Set *note* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil
note	the note

Since: 3.9.6

ags_sf2_synth_util_get_volume ()

```
gdouble
ags_sf2_synth_util_get_volume (AgsSF2SynthUtil *sf2_synth_util);
```

Get volume of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Returns

the volume

Since: 3.9.6

ags_sf2_synth_util_set_volume ()

```
void
ags_sf2_synth_util_set_volume (AgsSF2SynthUtil *sf2_synth_util,
                               gdouble volume);
```

Set *volume* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
volume	the volume	

Since: 3.9.6

ags_sf2_synth_util_get_frame_count ()

```
guint
ags_sf2_synth_util_get_frame_count (AgsSF2SynthUtil *sf2_synth_util);
```

Get frame count of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the frame count

Since: 3.9.6

ags_sf2_synth_util_set_frame_count ()

```
void
ags_sf2_synth_util_set_frame_count (AgsSF2SynthUtil *sf2_synth_util,
                                    guint frame_count);
```

Set *frame_count* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
frame_count	the frame count	

Since: 3.9.6

ags_sf2_synth_util_get_offset ()

```
guint
ags_sf2_synth_util_get_offset (AgsSF2SynthUtil *sf2_synth_util);
```

Get offset of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the offset

Since: 3.9.6

ags_sf2_synth_util_set_offset ()

```
void
ags_sf2_synth_util_set_offset (AgsSF2SynthUtil *sf2_synth_util,
                               guint offset);
```

Set *offset* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
offset	the offset	

Since: 3.9.6

ags_sf2_synth_util_get_loop_mode ()

```
guint
ags_sf2_synth_util_get_loop_mode (AgsSF2SynthUtil *sf2_synth_util);
```

Get loop mode of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the loop mode

Since: 3.9.6

ags_sf2_synth_util_set_loop_mode ()

```
void
ags_sf2_synth_util_set_loop_mode (AgsSF2SynthUtil *sf2_synth_util,
                                  guint loop_mode);
```

Set *loop_mode* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
loop_mode	the loop mode	

Since: 3.9.6

ags_sf2_synth_util_get_loop_start ()

```
guint
ags_sf2_synth_util_get_loop_start (AgsSF2SynthUtil *sf2_synth_util);
```

Get loop start of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the loop start

Since: 3.9.6

ags_sf2_synth_util_set_loop_start ()

```
void
ags_sf2_synth_util_set_loop_start (AgsSF2SynthUtil *sf2_synth_util,
                                   guint loop_start);
```

Set *loop_start* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
loop_start	the loop start	

Since: 3.9.6

ags_sf2_synth_util_get_loop_end ()

```
guint
ags_sf2_synth_util_get_loop_end (AgsSF2SynthUtil *sf2_synth_util);
```

Get loop end of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the loop end

Since: 3.9.6

ags_sf2_synth_util_set_loop_end ()

```
void
ags_sf2_synth_util_set_loop_end (AgsSF2SynthUtil *sf2_synth_util,
                                 guint loop_end);
```

Set *loop_end* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
loop_end	the loop end	

Since: 3.9.6

ags_sf2_synth_util_get_resample_util ()

```
AgsResampleUtil~*
ags_sf2_synth_util_get_resample_util (AgsSF2SynthUtil *sf2_synth_util);
```

Get resample util of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the resample util.

[transfer none]

Since: 3.9.6

ags_sf2_synth_util_set_resample_util ()

```
void
ags_sf2_synth_util_set_resample_util (AgsSF2SynthUtil *sf2_synth_util,
                                      AgsResampleUtil *resample_util);
```

Set *resample_util* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
resample_util	the resample util.	*[transfer full]*

Since: 3.9.6

ags_sf2_synth_util_get_pitch_type ()

```
GType
ags_sf2_synth_util_get_pitch_type (AgsSF2SynthUtil *sf2_synth_util);
```

Get pitch type of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the pitch type

Since: 4.0.0

ags_sf2_synth_util_set_pitch_type ()

```
void
ags_sf2_synth_util_set_pitch_type (AgsSF2SynthUtil *sf2_synth_util,
                                   GType pitch_type);
```

Set *pitch_type* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
pitch_type	the pitch type	

Since: 4.0.0

ags_sf2_synth_util_get_pitch_util ()

```
gpointer
ags_sf2_synth_util_get_pitch_util (AgsSF2SynthUtil *sf2_synth_util);
```

Get pitch util of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	

Returns

the pitch util.

[transfer none]

Since: 4.0.0

ags_sf2_synth_util_set_pitch_util ()

```
void
ags_sf2_synth_util_set_pitch_util (AgsSF2SynthUtil *sf2_synth_util,
                                   gpointer pitch_util);
```

Set *pitch_util* of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
pitch_util	the pitch util.	*[transfer full]*

Since: 4.0.0

ags_sf2_synth_util_load_instrument ()

```
void
ags_sf2_synth_util_load_instrument (AgsSF2SynthUtil *sf2_synth_util,
                                    gchar *preset,
                                    gchar *instrument);
```

Load instrument of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
preset	the preset	
instrument	the instrument	

Since: 4.0.0

ags_sf2_synth_util_load_midi_locale ()

```
void
ags_sf2_synth_util_load_midi_locale (AgsSF2SynthUtil *sf2_synth_util,
                                     gint bank,
                                     gint program);
```

Load midi locale of *sf2_synth_util* .

Parameters

sf2_synth_util	the AgsSF2SynthUtil	
bank	the bank	
program	the program	

Since: 3.16.0

ags_sf2_synth_util_compute_s8 ()

```
void
ags_sf2_synth_util_compute_s8 (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of signed 8 bit data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_s16 ()

```
void
ags_sf2_synth_util_compute_s16 (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of signed 16 bit data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_s24 ()

```
void
ags_sf2_synth_util_compute_s24 (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of signed 24 bit data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_s32 ()

```
void
ags_sf2_synth_util_compute_s32 (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of signed 32 bit data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_s64 ()

```
void
ags_sf2_synth_util_compute_s64 (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of signed 64 bit data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_float ()

```
void
ags_sf2_synth_util_compute_float (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of floating point data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_double ()

```
void
ags_sf2_synth_util_compute_double (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of double precision floating point data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute_complex ()

```
void
ags_sf2_synth_util_compute_complex (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth of complex data.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_compute ()

```
void
ags_sf2_synth_util_compute (AgsSF2SynthUtil *sf2_synth_util);
```

Compute Soundfont2 synth.

Parameters

sf2_synth_util	the AgsSF2SynthUtil

Since: 3.9.6

ags_sf2_synth_util_midi_locale_find_sample_near_midi_key ()

```
AgsIpatchSample~*
ags_sf2_synth_util_midi_locale_find_sample_near_midi_key
                                (AgsIpatch *ipatch,
                                 gint bank,
                                 gint program,
                                 gint midi_key,
                                 gchar **preset,
                                 gchar **instrument,
                                 gchar **sample);
```

Find sample near MIDI key.

Parameters

ipatch	the AgsIpatch	
bank	the bank	
program	the program	
midi_key	the MIDI key	
preset	the preset.	*[out]*
instrument	the instrument.	*[out]*
sample	the sample.	*[out]*

Returns

the AgsIpatchSample.

[transfer full]

Since: 3.4.0

AGS_SF2_SYNTH_UTIL()

```
#define AGS_SF2_SYNTH_UTIL(ptr) ((AgsSF2SynthUtil *)(ptr))
```

ags_sf2_synth_util_get_type ()

```
GType
ags_sf2_synth_util_get_type (void);
```

ags_sf2_synth_util_loop_mode_get_type ()

```
GType
ags_sf2_synth_util_loop_mode_get_type ();
```

Types and Values

enum AgsSF2SynthUtilFlags

Members

AGS_SF2_SYNTH_UTIL_COMPUTE_INSTRUMENT	
AGS_SF2_SYNTH_UTIL_COMPUTE_MIDI_LOCALE	

enum AgsSF2SynthUtilLoopMode

Enum values to specify loop mode.

Members

AGS_SF2_SYNTH_UTIL_LOOP_NONE	loop none
AGS_SF2_SYNTH_UTIL_LOOP_STANDARD	loop standard
AGS_SF2_SYNTH_UTIL_LOOP_RELEASE	loop release
AGS_SF2_SYNTH_UTIL_LOOP_PINGPONG	loop pingpong

AGS_TYPE_SF2_SYNTH_UTIL

```
#define AGS_TYPE_SF2_SYNTH_UTIL         (ags_sf2_synth_util_get_type())
```

struct AgsSF2SynthUtil

```
struct AgsSF2SynthUtil {
  guint flags;

  AgsAudioContainer *sf2_file;

  guint sf2_sample_count;
  IpatchSample **sf2_sample_arr;
  gint **sf2_note_range;

  gint sf2_loop_mode[128];
  gint sf2_loop_start[128];
  gint sf2_loop_end[128];

  guint sf2_orig_buffer_length[128];
  gpointer *sf2_orig_buffer;

  guint sf2_resampled_buffer_length[128];
  gpointer *sf2_resampled_buffer;

  gpointer source;
  guint source_stride;

  gpointer sample_buffer;
  gpointer im_buffer;

  guint buffer_length;
  guint format;
  guint samplerate;
  guint orig_samplerate;

  gchar *preset;
  gchar *instrument;
  gchar *sample;

  gint bank;
  gint program;

  gint midi_key;

  gdouble note;
  gdouble volume;

  guint frame_count;
  guint offset;

  guint loop_mode;

  gint loop_start;
  gint loop_end;

  AgsResampleUtil *resample_util;

  GType pitch_type;

  gpointer pitch_util;

  AgsVolumeUtil *volume_util;
};
```

5.23 AgsSFZSynthUtil

AgsSFZSynthUtil — SFZ synth util

Stability Level

Stable, unless otherwise indicated

Functions

AgsSFZSynthUtil *	ags_sfz_synth_util_alloc ()
gpointer	ags_sfz_synth_util_boxed_copy ()
void	ags_sfz_synth_util_free ()
AgsAudioContainer *	ags_sfz_synth_util_get_sfz_file ()
void	ags_sfz_synth_util_set_sfz_file ()
gpointer	ags_sfz_synth_util_get_source ()
void	ags_sfz_synth_util_set_source ()
guint	ags_sfz_synth_util_get_source_stride ()
void	ags_sfz_synth_util_set_source_stride ()
guint	ags_sfz_synth_util_get_buffer_length ()
void	ags_sfz_synth_util_set_buffer_length ()
guint	ags_sfz_synth_util_get_format ()
void	ags_sfz_synth_util_set_format ()
guint	ags_sfz_synth_util_get_samplerate ()
void	ags_sfz_synth_util_set_samplerate ()
gint	ags_sfz_synth_util_get_midi_key ()
void	ags_sfz_synth_util_set_midi_key ()
gdouble	ags_sfz_synth_util_get_note ()
void	ags_sfz_synth_util_set_note ()
gdouble	ags_sfz_synth_util_get_volume ()
void	ags_sfz_synth_util_set_volume ()
guint	ags_sfz_synth_util_get_frame_count ()
void	ags_sfz_synth_util_set_frame_count ()
guint	ags_sfz_synth_util_get_offset ()
void	ags_sfz_synth_util_set_offset ()
guint	ags_sfz_synth_util_get_loop_mode ()
void	ags_sfz_synth_util_set_loop_mode ()
guint	ags_sfz_synth_util_get_loop_start ()
void	ags_sfz_synth_util_set_loop_start ()
guint	ags_sfz_synth_util_get_loop_end ()
void	ags_sfz_synth_util_set_loop_end ()
GType	ags_sfz_synth_util_get_pitch_type ()
void	ags_sfz_synth_util_set_pitch_type ()
gpointer	ags_sfz_synth_util_get_pitch_util ()
void	ags_sfz_synth_util_set_pitch_util ()
void	ags_sfz_synth_util_load_instrument ()
void	ags_sfz_synth_util_compute_s8 ()
void	ags_sfz_synth_util_compute_s16 ()
void	ags_sfz_synth_util_compute_s24 ()
void	ags_sfz_synth_util_compute_s32 ()
void	ags_sfz_synth_util_compute_s64 ()
void	ags_sfz_synth_util_compute_float ()
void	ags_sfz_synth_util_compute_double ()
void	ags_sfz_synth_util_compute_complex ()

void	ags_sfz_synth_util_compute ()
#define	AGS_SFZ_SYNTH_UTIL()
GType	ags_sfz_synth_util_get_type ()
GType	ags_sfz_synth_util_loop_mode_get_type ()

Types and Values

enum	AgsSFZSynthUtilLoopMode
#define	AGS_TYPE_SFZ_SYNTH_UTIL
struct	AgsSFZSynthUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsSFZSynthUtil
    GEnum
    &#x2570;&#x2500;&#x2500; AgsSFZSynthUtilLoopMode
```

Includes

```
#include <ags/audio/ags_sfz_synth_util.h>
```

Description

Utility functions to compute SFZ synths.

Functions

ags_sfz_synth_util_alloc ()

```
AgsSFZSynthUtil~*
ags_sfz_synth_util_alloc ();
```

Allocate AgsSFZSynthUtil.

Returns

the newly allocated AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_boxed_copy ()

```
gpointer
ags_sfz_synth_util_boxed_copy (AgsSFZSynthUtil *ptr);
```

Copy AgsSFZSynthUtil.

Parameters

ptr	the AgsSFZSynthUtil

Returns

the newly allocated AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_free ()

```
void
ags_sfz_synth_util_free (AgsSFZSynthUtil *ptr);
```

Free AgsSFZSynthUtil.

Parameters

ptr	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_get_sfz_file ()

```
AgsAudioContainer~*
ags_sfz_synth_util_get_sfz_file (AgsSFZSynthUtil *sfz_synth_util);
```

Get SFZ file of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the SFZ file

Since: 4.0.0

ags_sfz_synth_util_set_sfz_file ()

```
void
ags_sfz_synth_util_set_sfz_file (AgsSFZSynthUtil *sfz_synth_util,
                                 AgsAudioContainer *sfz_file);
```

Set SFZ file of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
sfz_file	the SFZ file	

Since: 4.0.0

ags_sfz_synth_util_get_source ()

```
gpointer
ags_sfz_synth_util_get_source (AgsSFZSynthUtil *sfz_synth_util);
```

Get source buffer of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the source buffer

Since: 3.9.6

ags_sfz_synth_util_set_source ()

```
void
ags_sfz_synth_util_set_source (AgsSFZSynthUtil *sfz_synth_util,
                               gpointer source);
```

Set *source* buffer of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
source	the source buffer	

Since: 3.9.6

ags_sfz_synth_util_get_source_stride ()

```
guint
ags_sfz_synth_util_get_source_stride (AgsSFZSynthUtil *sfz_synth_util);
```

Get source stride of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the source buffer stride

Since: 3.9.6

ags_sfz_synth_util_set_source_stride ()

```
void
ags_sfz_synth_util_set_source_stride (AgsSFZSynthUtil *sfz_synth_util,
                                      guint source_stride);
```

Set *source* stride of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_sfz_synth_util_get_buffer_length ()

```
guint
ags_sfz_synth_util_get_buffer_length (AgsSFZSynthUtil *sfz_synth_util);
```

Get buffer length of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the buffer length

Since: 3.9.6

ags_sfz_synth_util_set_buffer_length ()

```
void
ags_sfz_synth_util_set_buffer_length (AgsSFZSynthUtil *sfz_synth_util,
                                      guint buffer_length);
```

Set *buffer_length* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_sfz_synth_util_get_format ()

```
guint
ags_sfz_synth_util_get_format (AgsSFZSynthUtil *sfz_synth_util);
```

Get format of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the format

Since: 3.9.6

ags_sfz_synth_util_set_format ()

```
void
ags_sfz_synth_util_set_format (AgsSFZSynthUtil *sfz_synth_util,
                               guint format);
```

Set *format* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil
format	the format

Since: 3.9.6

ags_sfz_synth_util_get_samplerate ()

```
guint
ags_sfz_synth_util_get_samplerate (AgsSFZSynthUtil *sfz_synth_util);
```

Get samplerate of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the samplerate

Since: 3.9.6

ags_sfz_synth_util_set_samplerate ()

```
void
ags_sfz_synth_util_set_samplerate (AgsSFZSynthUtil *sfz_synth_util,
                                   guint samplerate);
```

Set *samplerate* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_sfz_synth_util_get_midi_key ()

```
gint
ags_sfz_synth_util_get_midi_key (AgsSFZSynthUtil *sfz_synth_util);
```

Get midi_key of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the midi key

Since: 3.17.0

ags_sfz_synth_util_set_midi_key ()

```
void
ags_sfz_synth_util_set_midi_key (AgsSFZSynthUtil *sfz_synth_util,
                                 gint midi_key);
```

Set *midi_key* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
midi_key	the midi key	

Since: 3.17.0

ags_sfz_synth_util_get_note ()

```
gdouble
ags_sfz_synth_util_get_note (AgsSFZSynthUtil *sfz_synth_util);
```

Get note of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the note

Since: 3.9.6

ags_sfz_synth_util_set_note ()

```
void
ags_sfz_synth_util_set_note (AgsSFZSynthUtil *sfz_synth_util,
                             gdouble note);
```

Set *note* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
note	the note	

Since: 3.9.6

ags_sfz_synth_util_get_volume ()

```
gdouble
ags_sfz_synth_util_get_volume (AgsSFZSynthUtil *sfz_synth_util);
```

Get volume of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the volume

Since: 3.9.6

ags_sfz_synth_util_set_volume ()

```
void
ags_sfz_synth_util_set_volume (AgsSFZSynthUtil *sfz_synth_util,
                               gdouble volume);
```

Set *volume* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
volume	the volume	

Since: 3.9.6

ags_sfz_synth_util_get_frame_count ()

```
guint
ags_sfz_synth_util_get_frame_count (AgsSFZSynthUtil *sfz_synth_util);
```

Get frame count of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the frame count

Since: 3.9.6

ags_sfz_synth_util_set_frame_count ()

```
void
ags_sfz_synth_util_set_frame_count (AgsSFZSynthUtil *sfz_synth_util,
                                    guint frame_count);
```

Set *frame_count* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil
frame_count	the frame count

Since: 3.9.6

ags_sfz_synth_util_get_offset ()

```
guint
ags_sfz_synth_util_get_offset (AgsSFZSynthUtil *sfz_synth_util);
```

Get offset of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the offset

Since: 3.9.6

ags_sfz_synth_util_set_offset ()

```
void
ags_sfz_synth_util_set_offset (AgsSFZSynthUtil *sfz_synth_util,
                               guint offset);
```

Set *offset* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
offset	the offset	

Since: 3.9.6

ags_sfz_synth_util_get_loop_mode ()

```
guint
ags_sfz_synth_util_get_loop_mode (AgsSFZSynthUtil *sfz_synth_util);
```

Get loop mode of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the loop mode

Since: 3.9.6

ags_sfz_synth_util_set_loop_mode ()

```
void
ags_sfz_synth_util_set_loop_mode (AgsSFZSynthUtil *sfz_synth_util,
                                  guint loop_mode);
```

Set *loop_mode* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
loop_mode	the loop mode	

Since: 3.9.6

ags_sfz_synth_util_get_loop_start ()

```
guint
ags_sfz_synth_util_get_loop_start (AgsSFZSynthUtil *sfz_synth_util);
```

Get loop start of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the loop start

Since: 3.9.6

ags_sfz_synth_util_set_loop_start ()

```
void
ags_sfz_synth_util_set_loop_start (AgsSFZSynthUtil *sfz_synth_util,
                                   guint loop_start);
```

Set *loop_start* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil
loop_start	the loop start

Since: 3.9.6

ags_sfz_synth_util_get_loop_end ()

```
guint
ags_sfz_synth_util_get_loop_end (AgsSFZSynthUtil *sfz_synth_util);
```

Get loop end of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Returns

the loop end

Since: 3.9.6

ags_sfz_synth_util_set_loop_end ()

```
void
ags_sfz_synth_util_set_loop_end (AgsSFZSynthUtil *sfz_synth_util,
                                 guint loop_end);
```

Set *loop_end* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
loop_end	the loop end	

Since: 3.9.6

ags_sfz_synth_util_get_pitch_type ()

```
GType
ags_sfz_synth_util_get_pitch_type (AgsSFZSynthUtil *sfz_synth_util);
```

Get pitch type of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the pitch type

Since: 4.0.0

ags_sfz_synth_util_set_pitch_type ()

```
void
ags_sfz_synth_util_set_pitch_type (AgsSFZSynthUtil *sfz_synth_util,
                                   GType pitch_type);
```

Set *pitch_type* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
pitch_type	the pitch type	

Since: 4.0.0

ags_sfz_synth_util_get_pitch_util ()

```
gpointer
ags_sfz_synth_util_get_pitch_util (AgsSFZSynthUtil *sfz_synth_util);
```

Get pitch util of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Returns

the pitch util.

[transfer none]

Since: 4.0.0

ags_sfz_synth_util_set_pitch_util ()

```
void
ags_sfz_synth_util_set_pitch_util (AgsSFZSynthUtil *sfz_synth_util,
                                   gpointer generic_pitch_util);
```

Set *pitch_util* of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	
pitch_util	the pitch util.	*[transfer full]*

Since: 4.0.0

ags_sfz_synth_util_load_instrument ()

```
void
ags_sfz_synth_util_load_instrument (AgsSFZSynthUtil *sfz_synth_util);
```

Load instrument of *sfz_synth_util* .

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Since: 3.17.0

ags_sfz_synth_util_compute_s8 ()

```
void
ags_sfz_synth_util_compute_s8 (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of signed 8 bit data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil	

Since: 3.9.6

ags_sfz_synth_util_compute_s16 ()

```
void
ags_sfz_synth_util_compute_s16 (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of signed 16 bit data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute_s24 ()

```
void
ags_sfz_synth_util_compute_s24 (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of signed 24 bit data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute_s32 ()

```
void
ags_sfz_synth_util_compute_s32 (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of signed 32 bit data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute_s64 ()

```
void
ags_sfz_synth_util_compute_s64 (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of signed 64 bit data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute_float ()

```
void
ags_sfz_synth_util_compute_float (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of floating point data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute_double ()

```
void
ags_sfz_synth_util_compute_double (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of double precision floating point data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute_complex ()

```
void
ags_sfz_synth_util_compute_complex (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth of complex data.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

ags_sfz_synth_util_compute ()

```
void
ags_sfz_synth_util_compute (AgsSFZSynthUtil *sfz_synth_util);
```

Compute SFZ synth.

Parameters

sfz_synth_util	the AgsSFZSynthUtil

Since: 3.9.6

AGS_SFZ_SYNTH_UTIL()

```
#define AGS_SFZ_SYNTH_UTIL(ptr) ((AgsSFZSynthUtil *)(ptr))
```

ags_sfz_synth_util_get_type ()

```
GType
ags_sfz_synth_util_get_type (void);
```

ags_sfz_synth_util_loop_mode_get_type ()

```
GType
ags_sfz_synth_util_loop_mode_get_type ();
```

Types and Values

enum AgsSFZSynthUtilLoopMode

Enum values to specify loop mode.

Members

AGS_SFZ_SYNTH_UTIL_LOOP_NONE	loop none
AGS_SFZ_SYNTH_UTIL_LOOP_STANDARD	loop standard
AGS_SFZ_SYNTH_UTIL_LOOP_RELEASE	loop release
AGS_SFZ_SYNTH_UTIL_LOOP_PINGPONG	loop pingpong

AGS_TYPE_SFZ_SYNTH_UTIL

```
#define AGS_TYPE_SFZ_SYNTH_UTIL         (ags_sfz_synth_util_get_type())
```

struct AgsSFZSynthUtil

```
struct AgsSFZSynthUtil {
  guint flags;

  AgsAudioContainer *sfz_file;

  guint sfz_sample_count;
  AgsSFZSample **sfz_sample_arr;
  gint **sfz_note_range;

  gint sfz_loop_mode[128];
  gint sfz_loop_start[128];
  gint sfz_loop_end[128];

  guint sfz_orig_buffer_length[128];
  gpointer *sfz_orig_buffer;

  guint sfz_resampled_buffer_length[128];
  gpointer *sfz_resampled_buffer;

  gpointer source;
  guint source_stride;

  gpointer sample_buffer;
  gpointer im_buffer;

  guint buffer_length;
  guint format;
  guint samplerate;
  guint orig_samplerate;

  gint midi_key;

  gdouble note;
  gdouble volume;

  guint frame_count;
  guint offset;

  guint loop_mode;

  guint loop_start;
  guint loop_end;

  AgsResampleUtil *resample_util;

  GType pitch_type;

  gpointer pitch_util;

  AgsVolumeUtil *volume_util;
};
```

5.24 AgsLFOSynthUtil

AgsLFOSynthUtil — low frequency oscillator synth util

Stability Level

Stable, unless otherwise indicated

Functions

AgsLFOSynthUtil *	ags_lfo_synth_util_alloc ()
gpointer	ags_lfo_synth_util_copy ()
void	ags_lfo_synth_util_free ()
gpointer	ags_lfo_synth_util_get_source ()
void	ags_lfo_synth_util_set_source ()
guint	ags_lfo_synth_util_get_source_stride ()
void	ags_lfo_synth_util_set_source_stride ()
guint	ags_lfo_synth_util_get_buffer_length ()
void	ags_lfo_synth_util_set_buffer_length ()
guint	ags_lfo_synth_util_get_format ()
void	ags_lfo_synth_util_set_format ()
guint	ags_lfo_synth_util_get_samplerate ()
void	ags_lfo_synth_util_set_samplerate ()
guint	ags_lfo_synth_util_get_lfo_synth_oscillator_mode ()
void	ags_lfo_synth_util_set_lfo_synth_oscillator_mode ()
gdouble	ags_lfo_synth_util_get_frequency ()
void	ags_lfo_synth_util_set_frequency ()
gdouble	ags_lfo_synth_util_get_phase ()
void	ags_lfo_synth_util_set_phase ()
gdouble	ags_lfo_synth_util_get_lfo_depth ()
void	ags_lfo_synth_util_set_lfo_depth ()
gdouble	ags_lfo_synth_util_get_tuning ()
void	ags_lfo_synth_util_set_tuning ()
guint	ags_lfo_synth_util_get_offset ()
void	ags_lfo_synth_util_set_offset ()
guint	ags_lfo_synth_util_get_frame_count ()
void	ags_lfo_synth_util_set_frame_count ()
void	ags_lfo_synth_util_compute_sin_s8 ()
void	ags_lfo_synth_util_compute_sin_s16 ()
void	ags_lfo_synth_util_compute_sin_s24 ()
void	ags_lfo_synth_util_compute_sin_s32 ()
void	ags_lfo_synth_util_compute_sin_s64 ()
void	ags_lfo_synth_util_compute_sin_float ()
void	ags_lfo_synth_util_compute_sin_double ()
void	ags_lfo_synth_util_compute_sin_complex ()
void	ags_lfo_synth_util_compute_sin ()
void	ags_lfo_synth_util_compute_sawtooth_s8 ()
void	ags_lfo_synth_util_compute_sawtooth_s16 ()
void	ags_lfo_synth_util_compute_sawtooth_s24 ()
void	ags_lfo_synth_util_compute_sawtooth_s32 ()
void	ags_lfo_synth_util_compute_sawtooth_s64 ()
void	ags_lfo_synth_util_compute_sawtooth_float ()
void	ags_lfo_synth_util_compute_sawtooth_double ()
void	ags_lfo_synth_util_compute_sawtooth_complex ()
void	ags_lfo_synth_util_compute_sawtooth ()
void	ags_lfo_synth_util_compute_triangle_s8 ()
void	ags_lfo_synth_util_compute_triangle_s16 ()
void	ags_lfo_synth_util_compute_triangle_s24 ()
void	ags_lfo_synth_util_compute_triangle_s32 ()

void	ags_lfo_synth_util_compute_triangle_s64 ()
void	ags_lfo_synth_util_compute_triangle_float ()
void	ags_lfo_synth_util_compute_triangle_double ()
void	ags_lfo_synth_util_compute_triangle_complex ()
void	ags_lfo_synth_util_compute_triangle ()
void	ags_lfo_synth_util_compute_square_s8 ()
void	ags_lfo_synth_util_compute_square_s16 ()
void	ags_lfo_synth_util_compute_square_s24 ()
void	ags_lfo_synth_util_compute_square_s32 ()
void	ags_lfo_synth_util_compute_square_s64 ()
void	ags_lfo_synth_util_compute_square_float ()
void	ags_lfo_synth_util_compute_square_double ()
void	ags_lfo_synth_util_compute_square_complex ()
void	ags_lfo_synth_util_compute_square ()
void	ags_lfo_synth_util_compute_impulse_s8 ()
void	ags_lfo_synth_util_compute_impulse_s16 ()
void	ags_lfo_synth_util_compute_impulse_s24 ()
void	ags_lfo_synth_util_compute_impulse_s32 ()
void	ags_lfo_synth_util_compute_impulse_s64 ()
void	ags_lfo_synth_util_compute_impulse_float ()
void	ags_lfo_synth_util_compute_impulse_double ()
void	ags_lfo_synth_util_compute_impulse_complex ()
void	ags_lfo_synth_util_compute_impulse ()
GType	ags_lfo_synth_util_get_type ()

Types and Values

#define	AGS_LFO_SYNTH_UTIL_DEFAULT_FREQUENCY
#define	AGS_LFO_SYNTH_UTIL_DEFAULT_LFO_DEPTH
#define	AGS_LFO_SYNTH_UTIL_DEFAULT_TUNING
#define	AGS_TYPE_LFO_SYNTH_UTIL
struct	AgsLFOSynthUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsLFOSynthUtil
```

Includes

```
#include <ags/audio/ags_lfo_synth_util.h>
```

Description

Utility functions to compute LFO synths.

Functions

ags_lfo_synth_util_alloc ()

```
AgsLFOSynthUtil~*
ags_lfo_synth_util_alloc ();
```

Allocate AgsLFOSynthUtil

Returns

a new AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_copy ()

```
gpointer
ags_lfo_synth_util_copy (AgsLFOSynthUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsLFOSynthUtil

Returns

a pointer of the new AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_free ()

```
void
ags_lfo_synth_util_free (AgsLFOSynthUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_get_source ()

```
gpointer
ags_lfo_synth_util_get_source (AgsLFOSynthUtil *lfo_synth_util);
```

Get source buffer of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Returns

the source buffer

Since: 4.0.0

ags_lfo_synth_util_set_source ()

```
void
ags_lfo_synth_util_set_source (AgsLFOSynthUtil *lfo_synth_util,
                               gpointer source);
```

Set *source* buffer of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
source	the source buffer	

Since: 4.0.0

ags_lfo_synth_util_get_source_stride ()

```
guint
ags_lfo_synth_util_get_source_stride (AgsLFOSynthUtil *lfo_synth_util);
```

Get source stride of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the source buffer stride

Since: 4.0.0

ags_lfo_synth_util_set_source_stride ()

```
void
ags_lfo_synth_util_set_source_stride (AgsLFOSynthUtil *lfo_synth_util,
                                      guint source_stride);
```

Set *source* stride of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
source_stride	the source buffer stride	

Since: 4.0.0

ags_lfo_synth_util_get_buffer_length ()

```
guint
ags_lfo_synth_util_get_buffer_length (AgsLFOSynthUtil *lfo_synth_util);
```

Get buffer length of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Returns

the buffer length

Since: 4.0.0

ags_lfo_synth_util_set_buffer_length ()

```
void
ags_lfo_synth_util_set_buffer_length (AgsLFOSynthUtil *lfo_synth_util,
                                      guint buffer_length);
```

Set *buffer_length* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil
buffer_length	the buffer length

Since: 4.0.0

ags_lfo_synth_util_get_format ()

```
guint
ags_lfo_synth_util_get_format (AgsLFOSynthUtil *lfo_synth_util);
```

Get audio buffer util format of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Returns

the audio buffer util format

Since: 3.14.0

ags_lfo_synth_util_set_format ()

```
void
ags_lfo_synth_util_set_format (AgsLFOSynthUtil *lfo_synth_util,
                               guint format);
```

Set *format* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
format	the audio buffer util format	

Since: 3.14.0

ags_lfo_synth_util_get_samplerate ()

```
guint
ags_lfo_synth_util_get_samplerate (AgsLFOSynthUtil *lfo_synth_util);
```

Get samplerate of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the samplerate

Since: 4.0.0

ags_lfo_synth_util_set_samplerate ()

```
void
ags_lfo_synth_util_set_samplerate (AgsLFOSynthUtil *lfo_synth_util,
                                   guint samplerate);
```

Set *samplerate* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
samplerate	the samplerate	

Since: 4.0.0

ags_lfo_synth_util_get_lfo_synth_oscillator_mode ()

```
guint
ags_lfo_synth_util_get_lfo_synth_oscillator_mode
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Get LFO synth oscillator mode of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the LFO synth oscillator mode

Since: 4.0.0

ags_lfo_synth_util_set_lfo_synth_oscillator_mode ()

```
void
ags_lfo_synth_util_set_lfo_synth_oscillator_mode
                                (AgsLFOSynthUtil *lfo_synth_util,
                                 guint lfo_synth_oscillator_mode);
```

Set *lfo_synth_oscillator_mode* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
lfo_synth_oscillator_mode	the LFO synth oscillator mode	

Since: 4.0.0

ags_lfo_synth_util_get_frequency ()

```
gdouble
ags_lfo_synth_util_get_frequency (AgsLFOSynthUtil *lfo_synth_util);
```

Get frequency of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the frequency

Since: 4.0.0

ags_lfo_synth_util_set_frequency ()

```
void
ags_lfo_synth_util_set_frequency (AgsLFOSynthUtil *lfo_synth_util,
                                  gdouble frequency);
```

Set *frequency* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
frequency	the frequency	

Since: 4.0.0

ags_lfo_synth_util_get_phase ()

```
gdouble
ags_lfo_synth_util_get_phase (AgsLFOSynthUtil *lfo_synth_util);
```

Get phase of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the phase

Since: 4.0.0

ags_lfo_synth_util_set_phase ()

```
void
ags_lfo_synth_util_set_phase (AgsLFOSynthUtil *lfo_synth_util,
                              gdouble phase);
```

Set *phase* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
phase	the phase	

Since: 4.0.0

ags_lfo_synth_util_get_lfo_depth ()

```
gdouble
ags_lfo_synth_util_get_lfo_depth (AgsLFOSynthUtil *lfo_synth_util);
```

Get LFO depth of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the LFO depth

Since: 4.0.0

ags_lfo_synth_util_set_lfo_depth ()

```
void
ags_lfo_synth_util_set_lfo_depth (AgsLFOSynthUtil *lfo_synth_util,
                                  gdouble lfo_depth);
```

Set *lfo_depth* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
lfo_depth	the LFO depth	

Since: 4.0.0

ags_lfo_synth_util_get_tuning ()

```
gdouble
ags_lfo_synth_util_get_tuning (AgsLFOSynthUtil *lfo_synth_util);
```

Get tuning of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Returns

the tuning

Since: 4.0.0

ags_lfo_synth_util_set_tuning ()

```
void
ags_lfo_synth_util_set_tuning (AgsLFOSynthUtil *lfo_synth_util,
                               gdouble tuning);
```

Set *tuning* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
tuning	the tuning	

Since: 4.0.0

ags_lfo_synth_util_get_offset ()

```
guint
ags_lfo_synth_util_get_offset (AgsLFOSynthUtil *lfo_synth_util);
```

Get offset of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Returns

the offset

Since: 4.0.0

ags_lfo_synth_util_set_offset ()

```
void
ags_lfo_synth_util_set_offset (AgsLFOSynthUtil *lfo_synth_util,
                               guint offset);
```

Set *offset* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil
offset	the offset

Since: 4.0.0

ags_lfo_synth_util_get_frame_count ()

```
guint
ags_lfo_synth_util_get_frame_count (AgsLFOSynthUtil *lfo_synth_util);
```

Get frame count of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Returns

the frame count

Since: 4.0.0

ags_lfo_synth_util_set_frame_count ()

```
void
ags_lfo_synth_util_set_frame_count (AgsLFOSynthUtil *lfo_synth_util,
                                    guint frame_count);
```

Set *frame_count* of *lfo_synth_util* .

Parameters

lfo_synth_util	the AgsLFOSynthUtil	
frame_count	the frame count	

Since: 4.0.0

ags_lfo_synth_util_compute_sin_s8 ()

```
void
ags_lfo_synth_util_compute_sin_s8 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of signed 8 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Since: 4.0.0

ags_lfo_synth_util_compute_sin_s16 ()

```
void
ags_lfo_synth_util_compute_sin_s16 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of signed 16 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil	

Since: 4.0.0

ags_lfo_synth_util_compute_sin_s24 ()

```
void
ags_lfo_synth_util_compute_sin_s24 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of signed 24 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sin_s32 ()

```
void
ags_lfo_synth_util_compute_sin_s32 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of signed 32 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sin_s64 ()

```
void
ags_lfo_synth_util_compute_sin_s64 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of signed 64 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sin_float ()

```
void
ags_lfo_synth_util_compute_sin_float (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of single precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sin_double ()

```
void
ags_lfo_synth_util_compute_sin_double (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of double precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sin_complex ()

```
void
ags_lfo_synth_util_compute_sin_complex
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sine synth of complex data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sin ()

```
void
ags_lfo_synth_util_compute_sin (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_s8 ()

```
void
ags_lfo_synth_util_compute_sawtooth_s8
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of signed 8 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_s16 ()

```
void
ags_lfo_synth_util_compute_sawtooth_s16
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of signed 16 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_s24 ()

```
void
ags_lfo_synth_util_compute_sawtooth_s24
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of signed 24 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_s32 ()

```
void
ags_lfo_synth_util_compute_sawtooth_s32
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of signed 32 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_s64 ()

```
void
ags_lfo_synth_util_compute_sawtooth_s64
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of signed 64 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_float ()

```
void
ags_lfo_synth_util_compute_sawtooth_float
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of single precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_double ()

```
void
ags_lfo_synth_util_compute_sawtooth_double
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of double precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth_complex ()

```
void
ags_lfo_synth_util_compute_sawtooth_complex
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth of complex data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_sawtooth ()

```
void
ags_lfo_synth_util_compute_sawtooth (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO sawtooth synth.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_s8 ()

```
void
ags_lfo_synth_util_compute_triangle_s8
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of signed 8 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_s16 ()

```
void
ags_lfo_synth_util_compute_triangle_s16
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of signed 16 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_s24 ()

```
void
ags_lfo_synth_util_compute_triangle_s24
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of signed 24 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_s32 ()

```
void
ags_lfo_synth_util_compute_triangle_s32
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of signed 32 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_s64 ()

```
void
ags_lfo_synth_util_compute_triangle_s64
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of signed 64 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_float ()

```
void
ags_lfo_synth_util_compute_triangle_float
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of single precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_double ()

```
void
ags_lfo_synth_util_compute_triangle_double
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of double precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle_complex ()

```
void
ags_lfo_synth_util_compute_triangle_complex
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO triangle synth of complex data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_triangle ()

```
void
ags_lfo_synth_util_compute_triangle (AgsLFOSynthUtil *lfo_synth_util);
```

ags_lfo_synth_util_compute_square_s8 ()

```
void
ags_lfo_synth_util_compute_square_s8 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of signed 8 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_s16 ()

```
void
ags_lfo_synth_util_compute_square_s16 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of signed 16 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_s24 ()

```
void
ags_lfo_synth_util_compute_square_s24 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of signed 24 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_s32 ()

```
void
ags_lfo_synth_util_compute_square_s32 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of signed 32 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_s64 ()

```
void
ags_lfo_synth_util_compute_square_s64 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of signed 64 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_float ()

```
void
ags_lfo_synth_util_compute_square_float
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of single precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_double ()

```
void
ags_lfo_synth_util_compute_square_double
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of double precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square_complex ()

```
void
ags_lfo_synth_util_compute_square_complex
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth of complex data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_square ()

```
void
ags_lfo_synth_util_compute_square (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO square synth.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_s8 ()

```
void
ags_lfo_synth_util_compute_impulse_s8 (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of signed 8 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_s16 ()

```
void
ags_lfo_synth_util_compute_impulse_s16
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of signed 16 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_s24 ()

```
void
ags_lfo_synth_util_compute_impulse_s24
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of signed 24 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_s32 ()

```
void
ags_lfo_synth_util_compute_impulse_s32
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of signed 32 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_s64 ()

```
void
ags_lfo_synth_util_compute_impulse_s64
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of signed 64 bit data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_float ()

```
void
ags_lfo_synth_util_compute_impulse_float
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of single precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_double ()

```
void
ags_lfo_synth_util_compute_impulse_double
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of double precision floating point data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse_complex ()

```
void
ags_lfo_synth_util_compute_impulse_complex
                                (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth of complex data.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_compute_impulse ()

```
void
ags_lfo_synth_util_compute_impulse (AgsLFOSynthUtil *lfo_synth_util);
```

Compute LFO impulse synth.

Parameters

lfo_synth_util	the AgsLFOSynthUtil

Since: 4.0.0

ags_lfo_synth_util_get_type ()

```
GType
ags_lfo_synth_util_get_type (void);
```

Types and Values

AGS_LFO_SYNTH_UTIL_DEFAULT_FREQUENCY

```
#define AGS_LFO_SYNTH_UTIL_DEFAULT_FREQUENCY (12.0)
```

AGS_LFO_SYNTH_UTIL_DEFAULT_LFO_DEPTH

```
#define AGS_LFO_SYNTH_UTIL_DEFAULT_LFO_DEPTH (0.0)
```

AGS_LFO_SYNTH_UTIL_DEFAULT_TUNING

```
#define AGS_LFO_SYNTH_UTIL_DEFAULT_TUNING (0.0)
```

AGS_TYPE_LFO_SYNTH_UTIL

```
#define AGS_TYPE_LFO_SYNTH_UTIL         (ags_lfo_synth_util_get_type())
```

struct AgsLFOSynthUtil

```
struct AgsLFOSynthUtil {
  gpointer source;
  guint source_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint lfo_synth_oscillator_mode;

  gdouble frequency;
  gdouble phase;

  gdouble lfo_depth;
  gdouble tuning;

  guint offset;
  guint frame_count;
};
```

5.25 AgsSynthGenerator

AgsSynthGenerator — generate wave-form

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SYNTH_GENERATOR_GET_OBJ_MUTEX()
GRecMutex *	ags_synth_generator_get_obj_mutex ()
guint	ags_synth_generator_get_samplerate ()
void	ags_synth_generator_set_samplerate ()
guint	ags_synth_generator_get_buffer_size ()
void	ags_synth_generator_set_buffer_size ()
guint	ags_synth_generator_get_format ()
void	ags_synth_generator_set_format ()
gdouble	ags_synth_generator_get_delay ()
void	ags_synth_generator_set_delay ()
guint	ags_synth_generator_get_frame_count ()
void	ags_synth_generator_set_frame_count ()
guint	ags_synth_generator_get_loop_start ()
void	ags_synth_generator_set_loop_start ()
guint	ags_synth_generator_get_loop_end ()
void	ags_synth_generator_set_loop_end ()

gboolean	ags_synth_generator_get_do_lfo ()
void	ags_synth_generator_set_do_lfo ()
guint	ags_synth_generator_get_oscillator ()
void	ags_synth_generator_set_oscillator ()
gdouble	ags_synth_generator_get_frequency ()
void	ags_synth_generator_set_frequency ()
gdouble	ags_synth_generator_get_phase ()
void	ags_synth_generator_set_phase ()
gdouble	ags_synth_generator_get_volume ()
void	ags_synth_generator_set_volume ()
gdouble	ags_synth_generator_get_lfo_depth ()
void	ags_synth_generator_set_lfo_depth ()
gdouble	ags_synth_generator_get_tuning ()
void	ags_synth_generator_set_tuning ()
gboolean	ags_synth_generator_get_do_fm_synth ()
void	ags_synth_generator_set_do_fm_synth ()
guint	ags_synth_generator_get_fm_lfo_oscillator ()
void	ags_synth_generator_set_fm_lfo_oscillator ()
gdouble	ags_synth_generator_get_fm_lfo_frequency ()
void	ags_synth_generator_set_fm_lfo_frequency ()
gdouble	ags_synth_generator_get_fm_lfo_depth ()
void	ags_synth_generator_set_fm_lfo_depth ()
gdouble	ags_synth_generator_get_fm_tuning ()
void	ags_synth_generator_set_fm_tuning ()
AgsTimestamp *	ags_synth_generator_get_timestamp ()
void	ags_synth_generator_set_timestamp ()
void	ags_synth_generator_compute ()
AgsSynthGenerator *	ags_synth_generator_new ()
#define	AGS_IS_SYNTH_GENERATOR()
#define	AGS_IS_SYNTH_GENERATOR_CLASS()
#define	AGS_SYNTH_GENERATOR()
#define	AGS_SYNTH_GENERATOR_CLASS()
#define	AGS_SYNTH_GENERATOR_GET_CLASS()
GType	ags_synth_generator_get_type ()

Properties

guint	attack	Read / Write
guint	buffer-size	Read / Write
double	delay	Read / Write
gboolean	do-fm-synth	Read / Write
gboolean	do-lfo	Read / Write
double	fm-lfo-depth	Read / Write
double	fm-lfo-frequency	Read / Write
guint	fm-lfo-oscillator	Read / Write
double	fm-tuning	Read / Write
guint	format	Read / Write
guint	frame-count	Read / Write
double	frequency	Read / Write
double	lfo-depth	Read / Write
guint	loop-end	Read / Write
guint	loop-start	Read / Write
guint	oscillator	Read / Write
double	phase	Read / Write
guint	samplerate	Read / Write

GObject *	timestamp	Read / Write
double	tuning	Read / Write
double	volume	Read / Write

Types and Values

#define	AGS_SYNTH_GENERATOR_DEFAULT_SAMPLERATE
#define	AGS_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE
#define	AGS_SYNTH_GENERATOR_DEFAULT_FORMAT
#define	AGS_SYNTH_GENERATOR_DEFAULT_OSCILLATOR
#define	AGS_SYNTH_GENERATOR_DEFAULT_FREQUENCY
#define	AGS_SYNTH_GENERATOR_DEFAULT_PHASE
#define	AGS_SYNTH_GENERATOR_DEFAULT_VOLUME
#define	AGS_SYNTH_GENERATOR_DEFAULT_LFO_DEPTH
#define	AGS_SYNTH_GENERATOR_DEFAULT_TUNING
#define	AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_FREQUENCY
#define	AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_DEPTH
#define	AGS_SYNTH_GENERATOR_DEFAULT_FM_TUNING
#define	AGS_TYPE_SYNTH_GENERATOR
struct	AgsSynthGenerator
struct	AgsSynthGeneratorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFunction
        &#x2570;&#x2500;&#x2500; AgsSynthGenerator
```

Includes

```
#include <ags/file/ags_synth_generator.h>
```

Description

The AgsSynthGenerator generates wave-forms with oscillators.

Functions

AGS_SYNTH_GENERATOR_GET_OBJ_MUTEX()

```
#define AGS_SYNTH_GENERATOR_GET_OBJ_MUTEX(obj) (&(((AgsSynthGenerator *) obj)->obj_mutex))
```

ags_synth_generator_get_obj_mutex ()

```
GRecMutex~*
ags_synth_generator_get_obj_mutex (AgsSynthGenerator *synth_generator);
```

Get object mutex.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the GRecMutex to lock *synth_generator*

Since: 3.1.0

ags_synth_generator_get_samplerate ()

```
guint
ags_synth_generator_get_samplerate (AgsSynthGenerator *synth_generator);
```

Gets samplerate.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the samplerate

Since: 3.1.0

ags_synth_generator_set_samplerate ()

```
void
ags_synth_generator_set_samplerate (AgsSynthGenerator *synth_generator,
                                    guint samplerate);
```

Set samplerate.

Parameters

synth_generator	the AgsSynthGenerator
samplerate	the samplerate

Since: 3.0.0

ags_synth_generator_get_buffer_size ()

```
guint
ags_synth_generator_get_buffer_size (AgsSynthGenerator *synth_generator);
```

Gets buffer size.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the buffer size

Since: 3.1.0

ags_synth_generator_set_buffer_size ()

```
void
ags_synth_generator_set_buffer_size (AgsSynthGenerator *synth_generator,
                                     guint buffer_size);
```

Set buffer size.

Parameters

synth_generator	the AgsSynthGenerator	
buffer_size	the buffer size	

Since: 3.0.0

ags_synth_generator_get_format ()

```
guint
ags_synth_generator_get_format (AgsSynthGenerator *synth_generator);
```

Gets format.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the format

Since: 3.1.0

ags_synth_generator_set_format ()

```
void
ags_synth_generator_set_format (AgsSynthGenerator *synth_generator,
                                guint format);
```

Set format.

Parameters

synth_generator	the AgsSynthGenerator	
format	the format	

Since: 3.0.0

ags_synth_generator_get_delay ()

```
gdouble
ags_synth_generator_get_delay (AgsSynthGenerator *synth_generator);
```

Gets delay.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the delay

Since: 3.1.0

ags_synth_generator_set_delay ()

```
void
ags_synth_generator_set_delay (AgsSynthGenerator *synth_generator,
                               gdouble delay);
```

Sets delay.

Parameters

synth_generator	the AgsSynthGenerator	
delay	the delay	

Since: 3.1.0

ags_synth_generator_get_frame_count ()

```
guint
ags_synth_generator_get_frame_count (AgsSynthGenerator *synth_generator);
```

Gets frame count.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the frame count

Since: 3.1.0

ags_synth_generator_set_frame_count ()

```
void
ags_synth_generator_set_frame_count (AgsSynthGenerator *synth_generator,
                                     guint frame_count);
```

Sets frame count.

Parameters

synth_generator	the AgsSynthGenerator	
frame_count	the frame count	

Since: 3.1.0

ags_synth_generator_get_loop_start ()

```
guint
ags_synth_generator_get_loop_start (AgsSynthGenerator *synth_generator);
```

Gets loop start.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the loop start

Since: 3.1.0

ags_synth_generator_set_loop_start ()

```
void
ags_synth_generator_set_loop_start (AgsSynthGenerator *synth_generator,
                                    guint loop_start);
```

Sets loop start.

Parameters

synth_generator	the AgsSynthGenerator	
loop_start	the loop start	

Since: 3.1.0

ags_synth_generator_get_loop_end ()

```
guint
ags_synth_generator_get_loop_end (AgsSynthGenerator *synth_generator);
```

Gets loop end.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the loop end

Since: 3.1.0

ags_synth_generator_set_loop_end ()

```
void
ags_synth_generator_set_loop_end (AgsSynthGenerator *synth_generator,
                                  guint loop_end);
```

Sets loop end.

Parameters

synth_generator	the AgsSynthGenerator
loop_end	the loop end

Since: 3.1.0

ags_synth_generator_get_do_lfo ()

```
gboolean
ags_synth_generator_get_do_lfo (AgsSynthGenerator *synth_generator);
```

Gets do lfo.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the do lfo

Since: 3.1.0

ags_synth_generator_set_do_lfo ()

```
void
ags_synth_generator_set_do_lfo (AgsSynthGenerator *synth_generator,
                                gboolean do_lfo);
```

Sets do lfo.

Parameters

synth_generator	the AgsSynthGenerator	
do_lfo	the do lfo	

Since: 3.1.0

ags_synth_generator_get_oscillator ()

```
guint
ags_synth_generator_get_oscillator (AgsSynthGenerator *synth_generator);
```

Gets oscillator.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the oscillator

Since: 3.1.0

ags_synth_generator_set_oscillator ()

```
void
ags_synth_generator_set_oscillator (AgsSynthGenerator *synth_generator,
                                    guint oscillator);
```

Sets oscillator.

Parameters

synth_generator	the AgsSynthGenerator	
oscillator	the oscillator	

Since: 3.1.0

ags_synth_generator_get_frequency ()

```
gdouble
ags_synth_generator_get_frequency (AgsSynthGenerator *synth_generator);
```

Gets frequency.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the frequency

Since: 3.1.0

ags_synth_generator_set_frequency ()

```
void
ags_synth_generator_set_frequency (AgsSynthGenerator *synth_generator,
                                   gdouble frequency);
```

Sets frequency.

Parameters

synth_generator	the AgsSynthGenerator
frequency	the frequency

Since: 3.1.0

ags_synth_generator_get_phase ()

```
gdouble
ags_synth_generator_get_phase (AgsSynthGenerator *synth_generator);
```

Gets phase.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the phase

Since: 3.1.0

ags_synth_generator_set_phase ()

```
void
ags_synth_generator_set_phase (AgsSynthGenerator *synth_generator,
                               gdouble phase);
```

Sets phase.

Parameters

synth_generator	the AgsSynthGenerator	
phase	the phase	

Since: 3.1.0

ags_synth_generator_get_volume ()

```
gdouble
ags_synth_generator_get_volume (AgsSynthGenerator *synth_generator);
```

Gets volume.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the volume

Since: 3.1.0

ags_synth_generator_set_volume ()

```
void
ags_synth_generator_set_volume (AgsSynthGenerator *synth_generator,
                                gdouble volume);
```

Sets volume.

Parameters

synth_generator	the AgsSynthGenerator	
volume	the volume	

Since: 3.1.0

ags_synth_generator_get_lfo_depth ()

```
gdouble
ags_synth_generator_get_lfo_depth (AgsSynthGenerator *synth_generator);
```

Gets LFO depth.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the LFO depth

Since: 3.1.0

ags_synth_generator_set_lfo_depth ()

```
void
ags_synth_generator_set_lfo_depth (AgsSynthGenerator *synth_generator,
                                   gdouble lfo_depth);
```

Sets LFO depth.

Parameters

synth_generator	the AgsSynthGenerator	
lfo_depth	the LFO depth	

Since: 3.1.0

ags_synth_generator_get_tuning ()

```
gdouble
ags_synth_generator_get_tuning (AgsSynthGenerator *synth_generator);
```

Gets tuning.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the tuning

Since: 3.1.0

ags_synth_generator_set_tuning ()

```
void
ags_synth_generator_set_tuning (AgsSynthGenerator *synth_generator,
                                gdouble tuning);
```

Sets tuning.

Parameters

synth_generator	the AgsSynthGenerator	
tuning	the tuning	

Since: 3.1.0

ags_synth_generator_get_do_fm_synth ()

```
gboolean
ags_synth_generator_get_do_fm_synth (AgsSynthGenerator *synth_generator);
```

Gets do FM synth.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the do FM synth

Since: 3.1.0

ags_synth_generator_set_do_fm_synth ()

```
void
ags_synth_generator_set_do_fm_synth (AgsSynthGenerator *synth_generator,
                                     gboolean do_fm_synth);
```

Sets do FM synth.

Parameters

synth_generator	the AgsSynthGenerator
do_fm_synth	the do FM synth

Since: 3.1.0

ags_synth_generator_get_fm_lfo_oscillator ()

```
guint
ags_synth_generator_get_fm_lfo_oscillator
                                (AgsSynthGenerator *synth_generator);
```

Gets FM LFO oscillator.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the FM LFO oscillator

Since: 3.1.0

ags_synth_generator_set_fm_lfo_oscillator ()

```
void
ags_synth_generator_set_fm_lfo_oscillator
                                (AgsSynthGenerator *synth_generator,
                                 guint fm_lfo_oscillator);
```

Sets FM LFO oscillator.

Parameters

synth_generator	the AgsSynthGenerator	
fm_lfo_oscillator	the FM LFO oscillator	

Since: 3.1.0

ags_synth_generator_get_fm_lfo_frequency ()

```
gdouble
ags_synth_generator_get_fm_lfo_frequency
                                (AgsSynthGenerator *synth_generator);
```

Gets FM LFO frequency.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the FM LFO frequency

Since: 3.1.0

ags_synth_generator_set_fm_lfo_frequency ()

```
void
ags_synth_generator_set_fm_lfo_frequency
                                (AgsSynthGenerator *synth_generator,
                                 gdouble fm_lfo_frequency);
```

Sets FM LFO frequency.

Parameters

synth_generator	the AgsSynthGenerator	
fm_lfo_frequency	the FM LFO frequency	

Since: 3.1.0

ags_synth_generator_get_fm_lfo_depth ()

```
gdouble
ags_synth_generator_get_fm_lfo_depth (AgsSynthGenerator *synth_generator);
```

Gets FM LFO depth.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the FM LFO depth

Since: 3.1.0

ags_synth_generator_set_fm_lfo_depth ()

```
void
ags_synth_generator_set_fm_lfo_depth (AgsSynthGenerator *synth_generator,
                                      gdouble fm_lfo_depth);
```

Sets FM LFO depth.

Parameters

synth_generator	the AgsSynthGenerator
fm_lfo_depth	the FM LFO depth

Since: 3.1.0

ags_synth_generator_get_fm_tuning ()

```
gdouble
ags_synth_generator_get_fm_tuning (AgsSynthGenerator *synth_generator);
```

Gets FM tuning.

Parameters

synth_generator	the AgsSynthGenerator

Returns

the FM tuning

Since: 3.1.0

ags_synth_generator_set_fm_tuning ()

```
void
ags_synth_generator_set_fm_tuning (AgsSynthGenerator *synth_generator,
                                   gdouble fm_tuning);
```

Sets FM tuning.

Parameters

synth_generator	the AgsSynthGenerator	
fm_tuning	the FM tuning	

Since: 3.1.0

ags_synth_generator_get_timestamp ()

```
AgsTimestamp~*
ags_synth_generator_get_timestamp (AgsSynthGenerator *synth_generator);
```

Get timestamp.

Parameters

synth_generator	the AgsSynthGenerator	

Returns

the AgsTimestamp.

[transfer full]

Since: 3.1.0

ags_synth_generator_set_timestamp ()

```
void
ags_synth_generator_set_timestamp (AgsSynthGenerator *synth_generator,
                                   AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

synth_generator	the AgsSynthGenerator	
timestamp	the AgsTimestamp	

Since: 3.1.0

ags_synth_generator_compute ()

```
void
ags_synth_generator_compute (AgsSynthGenerator *synth_generator,
                             GObject *audio_signal,
                             gdouble note);
```

Compute synth for *note* .

Parameters

synth_generator	the AgsSynthGenerator	
audio_signal	the AgsAudioSignal	
note	the note to compute	

Since: 3.0.0

ags_synth_generator_new ()

```
AgsSynthGenerator~*
ags_synth_generator_new ();
```

Creates an AgsSynthGenerator

Returns

a new AgsSynthGenerator

Since: 3.0.0

AGS_IS_SYNTH_GENERATOR()

```
#define AGS_IS_SYNTH_GENERATOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SYNTH_GENERATOR))
```

AGS_IS_SYNTH_GENERATOR_CLASS()

```
#define AGS_IS_SYNTH_GENERATOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SYNTH_GENERATOR))
```

AGS_SYNTH_GENERATOR()

```
#define AGS_SYNTH_GENERATOR(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SYNTH_GENERATOR, AgsSynthGenerator))
```

AGS_SYNTH_GENERATOR_CLASS()

```
#define AGS_SYNTH_GENERATOR_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SYNTH_GENERATOR, AgsSynthGeneratorClass))
```

AGS_SYNTH_GENERATOR_GET_CLASS()

```
#define AGS_SYNTH_GENERATOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SYNTH_GENERATOR, AgsSynthGeneratorClass))
```

ags_synth_generator_get_type ()

```
GType
ags_synth_generator_get_type ();
```

Types and Values

AGS_SYNTH_GENERATOR_DEFAULT_SAMPLERATE

```
#define AGS_SYNTH_GENERATOR_DEFAULT_SAMPLERATE ((guint) AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE

```
#define AGS_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE (AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE)
```

AGS_SYNTH_GENERATOR_DEFAULT_FORMAT

```
#define AGS_SYNTH_GENERATOR_DEFAULT_FORMAT (AGS_SOUNDCARD_DEFAULT_FORMAT)
```

AGS_SYNTH_GENERATOR_DEFAULT_OSCILLATOR

```
#define AGS_SYNTH_GENERATOR_DEFAULT_OSCILLATOR (AGS_SYNTH_OSCILLATOR_SIN)
```

AGS_SYNTH_GENERATOR_DEFAULT_FREQUENCY

```
#define AGS_SYNTH_GENERATOR_DEFAULT_FREQUENCY (440.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_PHASE

```
#define AGS_SYNTH_GENERATOR_DEFAULT_PHASE (0.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_VOLUME

```
#define AGS_SYNTH_GENERATOR_DEFAULT_VOLUME (1.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_LFO_DEPTH

```
#define AGS_SYNTH_GENERATOR_DEFAULT_LFO_DEPTH (1.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_TUNING

```
#define AGS_SYNTH_GENERATOR_DEFAULT_TUNING (0.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_FREQUENCY

```
#define AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_FREQUENCY (6.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_DEPTH

```
#define AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_DEPTH (1.0)
```

AGS_SYNTH_GENERATOR_DEFAULT_FM_TUNING

```
#define AGS_SYNTH_GENERATOR_DEFAULT_FM_TUNING (0.0)
```

AGS_TYPE_SYNTH_GENERATOR

```
#define AGS_TYPE_SYNTH_GENERATOR                (ags_synth_generator_get_type())
```

struct AgsSynthGenerator

```
struct AgsSynthGenerator;
```

struct AgsSynthGeneratorClass

```
struct AgsSynthGeneratorClass {
  AgsFunctionClass function;
};
```

Property Details

The "attack" property

```
  "attack"                    guint
```

The attack to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 1024

Since: 3.0.0

The "delay" property

```
"delay"                      double
```

The delay to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,65535]

Default value: 0

Since: 3.0.0

The "do-fm-synth" property

```
"do-fm-synth"                gboolean
```

If TRUE compute FM synth, otherwise not.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "do-lfo" property

```
"do-lfo"                     gboolean
```

If TRUE compute LFO amplification, otherwise not.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "fm-lfo-depth" property

```
"fm-lfo-depth"              double
```

The fm LFO depth to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,1]

Default value: 1

Since: 3.0.0

The "fm-lfo-frequency" property

```
"fm-lfo-frequency"          double
```

The fm LFO frequency to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0.001,27.5]

Default value: 6

Since: 3.0.0

The "fm-lfo-oscillator" property

```
"fm-lfo-oscillator"         guint
```

The frame count to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "fm-tuning" property

```
"fm-tuning"                 double
```

The fm tuning to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,1200]

Default value: 0

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The format to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "frame-count" property

```
"frame-count"               guint
```

The frame count to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "frequency" property

```
"frequency"                 double
```

The frequency to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,65535]

Default value: 440

Since: 3.0.0

The "lfo-depth" property

```
"lfo-depth"                 double
```

The LFO depth to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,1]

Default value: 1

Since: 3.0.0

The "loop-end" property

```
"loop-end"                  guint
```

The loop end to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "loop-start" property

```
"loop-start"                guint
```

The loop start to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "oscillator" property

```
"oscillator"                guint
```

The oscillator to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "phase" property

```
"phase"                     double
```

The phase to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,65535]

Default value: 0

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Default value: 48000

Since: 3.0.0

The "timestamp" property

```
"timestamp"                 GObject~*
```

The assigned timestamp.

Owner: AgsSynthGenerator

Flags: Read / Write

Since: 3.0.0

The "tuning" property

```
"tuning"                    double
```

The tuning to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,1200]

Default value: 0

Since: 3.0.0

The "volume" property

```
"volume"                    double
```

The volume to be used.

Owner: AgsSynthGenerator

Flags: Read / Write

Allowed values: [0,2]

Default value: 1

Since: 3.0.0

5.26 AgsSF2SynthGenerator

AgsSF2SynthGenerator — generate wave-form

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SF2_SYNTH_GENERATOR_GET_OBJ_MUTEX()
GRecMutex *	ags_sf2_synth_generator_get_obj_mutex ()
gboolean	ags_sf2_synth_generator_test_flags ()
void	ags_sf2_synth_generator_set_flags ()
void	ags_sf2_synth_generator_unset_flags ()
gchar *	ags_sf2_synth_generator_get_filename ()
void	ags_sf2_synth_generator_set_filename ()
gchar *	ags_sf2_synth_generator_get_preset ()
void	ags_sf2_synth_generator_set_preset ()
gchar *	ags_sf2_synth_generator_get_instrument ()
void	ags_sf2_synth_generator_set_instrument ()
gint	ags_sf2_synth_generator_get_bank ()
void	ags_sf2_synth_generator_set_bank ()
gint	ags_sf2_synth_generator_get_program ()
void	ags_sf2_synth_generator_set_program ()
guint	ags_sf2_synth_generator_get_samplerate ()
void	ags_sf2_synth_generator_set_samplerate ()
guint	ags_sf2_synth_generator_get_buffer_size ()
void	ags_sf2_synth_generator_set_buffer_size ()
guint	ags_sf2_synth_generator_get_format ()
void	ags_sf2_synth_generator_set_format ()
gchar *	ags_sf2_synth_generator_get_pitch_type ()
void	ags_sf2_synth_generator_set_pitch_type ()
gdouble	ags_sf2_synth_generator_get_delay ()
void	ags_sf2_synth_generator_set_delay ()
guint	ags_sf2_synth_generator_get_frame_count ()
void	ags_sf2_synth_generator_set_frame_count ()
guint	ags_sf2_synth_generator_get_loop_start ()
void	ags_sf2_synth_generator_set_loop_start ()
guint	ags_sf2_synth_generator_get_loop_end ()
void	ags_sf2_synth_generator_set_loop_end ()
gdouble	ags_sf2_synth_generator_get_base_key ()
void	ags_sf2_synth_generator_set_base_key ()
gdouble	ags_sf2_synth_generator_get_tuning ()
void	ags_sf2_synth_generator_set_tuning ()
AgsTimestamp *	ags_sf2_synth_generator_get_timestamp ()
void	ags_sf2_synth_generator_set_timestamp ()
void	ags_sf2_synth_generator_compute ()
void	ags_sf2_synth_generator_compute_instrument ()
void	ags_sf2_synth_generator_compute_midi_locale ()
AgsSF2SynthGenerator *	ags_sf2_synth_generator_new ()
#define	AGS_IS_SF2_SYNTH_GENERATOR()
#define	AGS_IS_SF2_SYNTH_GENERATOR_CLASS()
#define	AGS_SF2_SYNTH_GENERATOR()
#define	AGS_SF2_SYNTH_GENERATOR_CLASS()
#define	AGS_SF2_SYNTH_GENERATOR_GET_CLASS()
GType	ags_sf2_synth_generator_get_type ()

Properties

guint	attack	Read / Write
int	bank	Read / Write
double	base-key	Read / Write
guint	buffer-size	Read / Write
double	delay	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
guint	frame-count	Read / Write
char *	instrument	Read / Write
guint	loop-end	Read / Write
guint	loop-start	Read / Write
char *	pitch-type	Read / Write
char *	preset	Read / Write
int	program	Read / Write
guint	samplerate	Read / Write
double	tuning	Read / Write
double	volume	Read / Write

Types and Values

#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_BANK
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_PROGRAM
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_SAMPLERATE
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_FORMAT
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_VOLUME
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_BASE_KEY
#define	AGS_SF2_SYNTH_GENERATOR_DEFAULT_TUNING
enum	AgsSF2SynthGeneratorFlags
#define	AGS_TYPE_SF2_SYNTH_GENERATOR
struct	AgsSF2SynthGenerator
struct	AgsSF2SynthGeneratorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFunction
        &#x2570;&#x2500;&#x2500; AgsSF2SynthGenerator
```

Includes

```
#include <ags/file/ags_sf2_synth_generator.h>
```

Description

The AgsSF2SynthGenerator generates wave-forms with oscillators.

Functions

AGS_SF2_SYNTH_GENERATOR_GET_OBJ_MUTEX()

```
#define AGS_SF2_SYNTH_GENERATOR_GET_OBJ_MUTEX(obj) (&(((AgsSF2SynthGenerator *) obj)-> ↩
    obj_mutex))
```

ags_sf2_synth_generator_get_obj_mutex ()

```
GRecMutex~*
ags_sf2_synth_generator_get_obj_mutex (AgsSF2SynthGenerator *sf2_synth_generator);
```

Get object mutex.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the GRecMutex to lock *sf2_synth_generator*

Since: 3.4.0

ags_sf2_synth_generator_test_flags ()

```
gboolean
ags_sf2_synth_generator_test_flags (AgsSF2SynthGenerator *sf2_synth_generator,
                                    guint flags);
```

Test *flags* to be set on *sf2_synth_generator* .

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.4.0

ags_sf2_synth_generator_set_flags ()

```
void
ags_sf2_synth_generator_set_flags (AgsSF2SynthGenerator *sf2_synth_generator,
                                   guint flags);
```

Set *flags* on *sf2_synth_generator* .

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
flags	the flags	

Since: 3.4.0

ags_sf2_synth_generator_unset_flags ()

```
void
ags_sf2_synth_generator_unset_flags (AgsSF2SynthGenerator *sf2_synth_generator,
                                     guint flags);
```

Unset *flags* on *sf2_synth_generator* .

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
flags	the flags	

Since: 3.4.0

ags_sf2_synth_generator_get_filename ()

```
gchar~*
ags_sf2_synth_generator_get_filename (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets filename.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the filename

Since: 3.4.0

ags_sf2_synth_generator_set_filename ()

```
void
ags_sf2_synth_generator_set_filename (AgsSF2SynthGenerator *sf2_synth_generator,
                                      gchar *filename);
```

Set filename.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
filename	the filename	

Since: 3.4.0

ags_sf2_synth_generator_get_preset ()

```
gchar~*
ags_sf2_synth_generator_get_preset (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets preset.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the preset

Since: 3.4.0

ags_sf2_synth_generator_set_preset ()

```
void
ags_sf2_synth_generator_set_preset (AgsSF2SynthGenerator *sf2_synth_generator,
                                    gchar *preset);
```

Set preset.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
preset	the preset	

Since: 3.4.0

ags_sf2_synth_generator_get_instrument ()

```
gchar~*
ags_sf2_synth_generator_get_instrument
                                (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets instrument.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the instrument

Since: 3.4.0

ags_sf2_synth_generator_set_instrument ()

```
void
ags_sf2_synth_generator_set_instrument
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 gchar *instrument);
```

Set instrument.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
instrument	the instrument	

Since: 3.4.0

ags_sf2_synth_generator_get_bank ()

```
gint
ags_sf2_synth_generator_get_bank (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets bank.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the bank

Since: 3.4.0

ags_sf2_synth_generator_set_bank ()

```
void
ags_sf2_synth_generator_set_bank (AgsSF2SynthGenerator *sf2_synth_generator,
                                  gint bank);
```

Set buffer size.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
bank	the buffer size	

Since: 3.4.0

ags_sf2_synth_generator_get_program ()

```
gint
ags_sf2_synth_generator_get_program (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets program.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the program

Since: 3.4.0

ags_sf2_synth_generator_set_program ()

```
void
ags_sf2_synth_generator_set_program (AgsSF2SynthGenerator *sf2_synth_generator,
                                     gint program);
```

Set buffer size.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator
program	the buffer size

Since: 3.4.0

ags_sf2_synth_generator_get_samplerate ()

```
guint
ags_sf2_synth_generator_get_samplerate
                                (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets samplerate.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the samplerate

Since: 3.4.0

ags_sf2_synth_generator_set_samplerate ()

```
void
ags_sf2_synth_generator_set_samplerate
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 guint samplerate);
```

Set samplerate.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
samplerate	the samplerate	

Since: 3.4.0

ags_sf2_synth_generator_get_buffer_size ()

```
guint
ags_sf2_synth_generator_get_buffer_size
                                (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets buffer size.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the buffer size

Since: 3.4.0

ags_sf2_synth_generator_set_buffer_size ()

```
void
ags_sf2_synth_generator_set_buffer_size
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 guint buffer_size);
```

Set buffer size.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
buffer_size	the buffer size	

Since: 3.4.0

ags_sf2_synth_generator_get_format ()

```
guint
ags_sf2_synth_generator_get_format (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets format.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the format

Since: 3.4.0

ags_sf2_synth_generator_set_format ()

```
void
ags_sf2_synth_generator_set_format (AgsSF2SynthGenerator *sf2_synth_generator,
                                    guint format);
```

Set format.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator
format	the format

Since: 3.4.0

ags_sf2_synth_generator_get_pitch_type ()

```
gchar~*
ags_sf2_synth_generator_get_pitch_type
                                (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets pitch type.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the pitch type

Since: 3.9.0

ags_sf2_synth_generator_set_pitch_type ()

```
void
ags_sf2_synth_generator_set_pitch_type
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 gchar *pitch_type);
```

Set pitch type.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
pitch_type	the pitch type	

Since: 3.9.0

ags_sf2_synth_generator_get_delay ()

```
gdouble
ags_sf2_synth_generator_get_delay (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets delay.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the delay

Since: 3.4.0

ags_sf2_synth_generator_set_delay ()

```
void
ags_sf2_synth_generator_set_delay (AgsSF2SynthGenerator *sf2_synth_generator,
                                   gdouble delay);
```

Sets delay.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
delay	the delay	

Since: 3.4.0

ags_sf2_synth_generator_get_frame_count ()

```
guint
ags_sf2_synth_generator_get_frame_count
                                (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets frame count.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the frame count

Since: 3.4.0

ags_sf2_synth_generator_set_frame_count ()

```
void
ags_sf2_synth_generator_set_frame_count
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 guint frame_count);
```

Sets frame count.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator
frame_count	the frame count

Since: 3.4.0

ags_sf2_synth_generator_get_loop_start ()

```
guint
ags_sf2_synth_generator_get_loop_start
                                (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets loop start.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the loop start

Since: 3.4.0

ags_sf2_synth_generator_set_loop_start ()

```
void
ags_sf2_synth_generator_set_loop_start
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 guint loop_start);
```

Sets loop start.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
loop_start	the loop start	

Since: 3.4.0

ags_sf2_synth_generator_get_loop_end ()

```
guint
ags_sf2_synth_generator_get_loop_end (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets loop end.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the loop end

Since: 3.4.0

ags_sf2_synth_generator_set_loop_end ()

```
void
ags_sf2_synth_generator_set_loop_end (AgsSF2SynthGenerator *sf2_synth_generator,
                                      guint loop_end);
```

Sets loop end.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
loop_end	the loop end	

Since: 3.4.0

ags_sf2_synth_generator_get_base_key ()

```
gdouble
ags_sf2_synth_generator_get_base_key (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets base key.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the base key

Since: 3.4.0

ags_sf2_synth_generator_set_base_key ()

```
void
ags_sf2_synth_generator_set_base_key (AgsSF2SynthGenerator *sf2_synth_generator,
                                      gdouble base_key);
```

Sets base key.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator
base_key	the base key

Since: 3.4.0

ags_sf2_synth_generator_get_tuning ()

```
gdouble
ags_sf2_synth_generator_get_tuning (AgsSF2SynthGenerator *sf2_synth_generator);
```

Gets tuning.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator

Returns

the tuning

Since: 3.4.0

ags_sf2_synth_generator_set_tuning ()

```
void
ags_sf2_synth_generator_set_tuning (AgsSF2SynthGenerator *sf2_synth_generator,
                                    gdouble tuning);
```

Sets tuning.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
tuning	the tuning	

Since: 3.4.0

ags_sf2_synth_generator_get_timestamp ()

```
AgsTimestamp~*
ags_sf2_synth_generator_get_timestamp (AgsSF2SynthGenerator *sf2_synth_generator);
```

Get timestamp.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	

Returns

the AgsTimestamp.

[transfer full]

Since: 3.4.0

ags_sf2_synth_generator_set_timestamp ()

```
void
ags_sf2_synth_generator_set_timestamp (AgsSF2SynthGenerator *sf2_synth_generator,
                                       AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
timestamp	the AgsTimestamp	

Since: 3.4.0

ags_sf2_synth_generator_compute ()

```
void
ags_sf2_synth_generator_compute (AgsSF2SynthGenerator *sf2_synth_generator,
                                 GObject *audio_signal,
                                 gdouble note);
```

Compute SF2 synth for *note* .

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
audio_signal	the AgsAudioSignal	
note	the note to compute	

Since: 3.4.0

ags_sf2_synth_generator_compute_instrument ()

```
void
ags_sf2_synth_generator_compute_instrument
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 GObject *audio_signal,
                                 gchar *preset,
                                 gchar *instrument,
                                 gdouble note);
```

Compute instrument for *note* .

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
audio_signal	the AgsAudioSignal	
preset	the preset	
instrument	the instrument	
note	the note to compute	

Since: 3.4.0

ags_sf2_synth_generator_compute_midi_locale ()

```
void
ags_sf2_synth_generator_compute_midi_locale
                                (AgsSF2SynthGenerator *sf2_synth_generator,
                                 GObject *audio_signal,
                                 gint bank,
                                 gint program,
                                 gdouble note);
```

Compute synth for *note* .

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
audio_signal	the AgsAudioSignal	
bank	the bank	
program	the program	
note	the note to compute	

Since: 3.4.0

ags_sf2_synth_generator_new ()

```
AgsSF2SynthGenerator~*
ags_sf2_synth_generator_new ();
```

Creates an AgsSF2SynthGenerator

Returns

a new AgsSF2SynthGenerator

Since: 3.4.0

AGS_IS_SF2_SYNTH_GENERATOR()

```
#define AGS_IS_SF2_SYNTH_GENERATOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SF2_SYNTH_GENERATOR))
```

AGS_IS_SF2_SYNTH_GENERATOR_CLASS()

```
#define AGS_IS_SF2_SYNTH_GENERATOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SF2_SYNTH_GENERATOR))
```

AGS_SF2_SYNTH_GENERATOR()

```
#define AGS_SF2_SYNTH_GENERATOR(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SF2_SYNTH_GENERATOR, AgsSF2SynthGenerator))
```

AGS_SF2_SYNTH_GENERATOR_CLASS()

```
#define AGS_SF2_SYNTH_GENERATOR_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SF2_SYNTH_GENERATOR, AgsSF2SynthGeneratorClass))
```

AGS_SF2_SYNTH_GENERATOR_GET_CLASS()

```
#define AGS_SF2_SYNTH_GENERATOR_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SF2_SYNTH_GENERATOR, AgsSF2SynthGeneratorClass))
```

ags_sf2_synth_generator_get_type ()

```
GType
ags_sf2_synth_generator_get_type ();
```

Types and Values

AGS_SF2_SYNTH_GENERATOR_DEFAULT_BANK

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_BANK (0)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_PROGRAM

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_PROGRAM (0)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_SAMPLERATE

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_SAMPLERATE ((guint)  ↩
    AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE (AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_FORMAT

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_FORMAT (AGS_SOUNDCARD_DEFAULT_FORMAT)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_VOLUME

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_VOLUME (1.0)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE "ags-fluid-4th-order"
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_BASE_KEY

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_BASE_KEY (-48.0)
```

AGS_SF2_SYNTH_GENERATOR_DEFAULT_TUNING

```
#define AGS_SF2_SYNTH_GENERATOR_DEFAULT_TUNING (0.0)
```

enum AgsSF2SynthGeneratorFlags

Members

AGS_SF2_SYNTH_GENERATOR_COMPUTE_INSTRUMENT		
AGS_SF2_SYNTH_GENERATOR_COMPUTE_MIDI_LOCALE		

AGS_TYPE_SF2_SYNTH_GENERATOR

```
#define AGS_TYPE_SF2_SYNTH_GENERATOR                (ags_sf2_synth_generator_get_type())
```

struct AgsSF2SynthGenerator

```
struct AgsSF2SynthGenerator;
```

struct AgsSF2SynthGeneratorClass

```
struct AgsSF2SynthGeneratorClass {
  AgsFunctionClass function;
};
```

Property Details

The "attack" property

```
  "attack"                    guint
```

The attack to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.11

The "bank" property

```
  "bank"                      int
```

The bank to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.4.0

The "base-key" property

```
"base-key"                  double
```

The base key to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Allowed values: [-65535,65535]

Default value: -48

Since: 3.4.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 1024

Since: 3.4.0

The "delay" property

```
"delay"                     double
```

The delay to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Allowed values: [0,65535]

Default value: 0

Since: 3.4.11

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: NULL

Since: 3.4.0

The "format" property

```
"format"                    guint
```

The format to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 16

Since: 3.4.0

The "frame-count" property

```
"frame-count"               guint
```

The frame count to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "instrument" property

```
"instrument"                char~*
```

The assigned instrument.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: NULL

Since: 3.4.0

The "loop-end" property

```
"loop-end"                  guint
```

The loop end to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "loop-start" property

```
"loop-start"                guint
```

The loop start to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "pitch-type" property

```
"pitch-type"                char~*
```

The pitch type to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: "ags-fluid-4th-order"

Since: 3.9.0

The "preset" property

```
"preset"                    char~*
```

The assigned preset.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: NULL

Since: 3.4.0

The "program" property

```
"program"                   int
```

The program to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.4.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Default value: 48000

Since: 3.4.0

The "tuning" property

```
"tuning"                    double
```

The tuning to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Allowed values: [0,1200]

Default value: 0

Since: 3.4.0

The "volume" property

```
"volume"                    double
```

The volume to be used.

Owner: AgsSF2SynthGenerator

Flags: Read / Write

Allowed values: [0,2]

Default value: 1

Since: 3.4.11

5.27 AgsSFZSynthGenerator

AgsSFZSynthGenerator — generate wave-form

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_SYNTH_GENERATOR_GET_OBJ_MUTEX()
GRecMutex *	ags_sfz_synth_generator_get_obj_mutex ()
gchar *	ags_sfz_synth_generator_get_filename ()
void	ags_sfz_synth_generator_set_filename ()
guint	ags_sfz_synth_generator_get_samplerate ()
void	ags_sfz_synth_generator_set_samplerate ()
guint	ags_sfz_synth_generator_get_buffer_size ()
void	ags_sfz_synth_generator_set_buffer_size ()
guint	ags_sfz_synth_generator_get_format ()
void	ags_sfz_synth_generator_set_format ()
gchar *	ags_sfz_synth_generator_get_pitch_type ()
void	ags_sfz_synth_generator_set_pitch_type ()
gdouble	ags_sfz_synth_generator_get_delay ()
void	ags_sfz_synth_generator_set_delay ()

guint	ags_sfz_synth_generator_get_frame_count ()
void	ags_sfz_synth_generator_set_frame_count ()
guint	ags_sfz_synth_generator_get_loop_start ()
void	ags_sfz_synth_generator_set_loop_start ()
guint	ags_sfz_synth_generator_get_loop_end ()
void	ags_sfz_synth_generator_set_loop_end ()
gdouble	ags_sfz_synth_generator_get_base_key ()
void	ags_sfz_synth_generator_set_base_key ()
gdouble	ags_sfz_synth_generator_get_tuning ()
void	ags_sfz_synth_generator_set_tuning ()
AgsTimestamp *	ags_sfz_synth_generator_get_timestamp ()
void	ags_sfz_synth_generator_set_timestamp ()
void	ags_sfz_synth_generator_compute ()
AgsSFZSynthGenerator *	ags_sfz_synth_generator_new ()
#define	AGS_IS_SFZ_SYNTH_GENERATOR()
#define	AGS_IS_SFZ_SYNTH_GENERATOR_CLASS()
#define	AGS_SFZ_SYNTH_GENERATOR()
#define	AGS_SFZ_SYNTH_GENERATOR_CLASS()
#define	AGS_SFZ_SYNTH_GENERATOR_GET_CLASS()
GType	ags_sfz_synth_generator_get_type ()

Properties

guint	attack	Read / Write
double	base-key	Read / Write
guint	buffer-size	Read / Write
double	delay	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
guint	frame-count	Read / Write
guint	loop-end	Read / Write
guint	loop-start	Read / Write
char *	pitch-type	Read / Write
guint	samplerate	Read / Write
double	tuning	Read / Write
double	volume	Read / Write

Types and Values

#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_SAMPLERATE
#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE
#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_FORMAT
#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_VOLUME
#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE
#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BASE_KEY
#define	AGS_SFZ_SYNTH_GENERATOR_DEFAULT_TUNING
#define	AGS_TYPE_SFZ_SYNTH_GENERATOR
struct	AgsSFZSynthGenerator
struct	AgsSFZSynthGeneratorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFunction
        &#x2570;&#x2500;&#x2500; AgsSFZSynthGenerator
```

Includes

```
#include <ags/file/ags_sfz_synth_generator.h>
```

Description

The AgsSFZSynthGenerator generates wave-forms with oscillators.

Functions

AGS_SFZ_SYNTH_GENERATOR_GET_OBJ_MUTEX()

```
#define AGS_SFZ_SYNTH_GENERATOR_GET_OBJ_MUTEX(obj) (&(((AgsSFZSynthGenerator *) obj)-> ↩
    obj_mutex))
```

ags_sfz_synth_generator_get_obj_mutex ()

```
GRecMutex~*
ags_sfz_synth_generator_get_obj_mutex (AgsSFZSynthGenerator *sfz_synth_generator);
```

Get object mutex.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the GRecMutex to lock *sfz_synth_generator*

Since: 3.4.0

ags_sfz_synth_generator_get_filename ()

```
gchar~*
ags_sfz_synth_generator_get_filename (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets filename.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the filename

Since: 3.4.0

ags_sfz_synth_generator_set_filename ()

```
void
ags_sfz_synth_generator_set_filename (AgsSFZSynthGenerator *sfz_synth_generator,
                                      gchar *filename);
```

Set filename.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
filename	the filename	

Since: 3.4.0

ags_sfz_synth_generator_get_samplerate ()

```
guint
ags_sfz_synth_generator_get_samplerate
                                (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets samplerate.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	

Returns

the samplerate

Since: 3.4.0

ags_sfz_synth_generator_set_samplerate ()

```
void
ags_sfz_synth_generator_set_samplerate
                                (AgsSFZSynthGenerator *sfz_synth_generator,
                                 guint samplerate);
```

Set samplerate.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
samplerate	the samplerate	

Since: 3.4.0

ags_sfz_synth_generator_get_buffer_size ()

```
guint
ags_sfz_synth_generator_get_buffer_size
                                (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets buffer size.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	

Returns

the buffer size

Since: 3.4.0

ags_sfz_synth_generator_set_buffer_size ()

```
void
ags_sfz_synth_generator_set_buffer_size
                                (AgsSFZSynthGenerator *sfz_synth_generator,
                                 guint buffer_size);
```

Set buffer size.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
buffer_size	the buffer size	

Since: 3.4.0

ags_sfz_synth_generator_get_format ()

```
guint
ags_sfz_synth_generator_get_format (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets format.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the format

Since: 3.4.0

ags_sfz_synth_generator_set_format ()

```
void
ags_sfz_synth_generator_set_format (AgsSFZSynthGenerator *sfz_synth_generator,
                                    guint format);
```

Set format.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator
format	the format

Since: 3.4.0

ags_sfz_synth_generator_get_pitch_type ()

```
gchar~*
ags_sfz_synth_generator_get_pitch_type
                                (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets pitch type.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the pitch type

Since: 3.9.0

ags_sfz_synth_generator_set_pitch_type ()

```
void
ags_sfz_synth_generator_set_pitch_type
                                (AgsSFZSynthGenerator *sfz_synth_generator,
                                 gchar *pitch_type);
```

Set pitch type.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
pitch_type	the pitch type	

Since: 3.9.0

ags_sfz_synth_generator_get_delay ()

```
gdouble
ags_sfz_synth_generator_get_delay (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets delay.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	

Returns

the delay

Since: 3.4.0

ags_sfz_synth_generator_set_delay ()

```
void
ags_sfz_synth_generator_set_delay (AgsSFZSynthGenerator *sfz_synth_generator,
                                   gdouble delay);
```

Sets delay.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
delay	the delay	

Since: 3.4.0

ags_sfz_synth_generator_get_frame_count ()

```
guint
ags_sfz_synth_generator_get_frame_count
                                (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets frame count.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the frame count

Since: 3.4.0

ags_sfz_synth_generator_set_frame_count ()

```
void
ags_sfz_synth_generator_set_frame_count
                                (AgsSFZSynthGenerator *sfz_synth_generator,
                                 guint frame_count);
```

Sets frame count.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator
frame_count	the frame count

Since: 3.4.0

ags_sfz_synth_generator_get_loop_start ()

```
guint
ags_sfz_synth_generator_get_loop_start
                                (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets loop start.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the loop start

Since: 3.4.0

ags_sfz_synth_generator_set_loop_start ()

```
void
ags_sfz_synth_generator_set_loop_start
                                (AgsSFZSynthGenerator *sfz_synth_generator,
                                 guint loop_start);
```

Sets loop start.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
loop_start	the loop start	

Since: 3.4.0

ags_sfz_synth_generator_get_loop_end ()

```
guint
ags_sfz_synth_generator_get_loop_end (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets loop end.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	

Returns

the loop end

Since: 3.4.0

ags_sfz_synth_generator_set_loop_end ()

```
void
ags_sfz_synth_generator_set_loop_end (AgsSFZSynthGenerator *sfz_synth_generator,
                                      guint loop_end);
```

Sets loop end.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
loop_end	the loop end	

Since: 3.4.0

ags_sfz_synth_generator_get_base_key ()

```
gdouble
ags_sfz_synth_generator_get_base_key (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets base key.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	

Returns

the base key

Since: 3.4.0

ags_sfz_synth_generator_set_base_key ()

```
void
ags_sfz_synth_generator_set_base_key (AgsSFZSynthGenerator *sfz_synth_generator,
                                      gdouble base_key);
```

Sets base key.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
base_key	the base key	

Since: 3.4.0

ags_sfz_synth_generator_get_tuning ()

```
gdouble
ags_sfz_synth_generator_get_tuning (AgsSFZSynthGenerator *sfz_synth_generator);
```

Gets tuning.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	

Returns

the tuning

Since: 3.4.0

ags_sfz_synth_generator_set_tuning ()

```
void
ags_sfz_synth_generator_set_tuning (AgsSFZSynthGenerator *sfz_synth_generator,
                                    gdouble tuning);
```

Sets tuning.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
tuning	the tuning	

Since: 3.4.0

ags_sfz_synth_generator_get_timestamp ()

```
AgsTimestamp~*
ags_sfz_synth_generator_get_timestamp (AgsSFZSynthGenerator *sfz_synth_generator);
```

Get timestamp.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator

Returns

the AgsTimestamp.

[transfer full]

Since: 3.4.0

ags_sfz_synth_generator_set_timestamp ()

```
void
ags_sfz_synth_generator_set_timestamp (AgsSFZSynthGenerator *sfz_synth_generator,
                                       AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator
timestamp	the AgsTimestamp

Since: 3.4.0

ags_sfz_synth_generator_compute ()

```
void
ags_sfz_synth_generator_compute (AgsSFZSynthGenerator *sfz_synth_generator,
                                 GObject *audio_signal,
                                 gdouble note);
```

Compute instrument for `note` .

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator
audio_signal	the AgsAudioSignal
note	the note to compute

Since: 3.4.0

ags_sfz_synth_generator_new ()

```
AgsSFZSynthGenerator~*
ags_sfz_synth_generator_new ();
```

Creates an AgsSFZSynthGenerator

Returns

a new AgsSFZSynthGenerator

Since: 3.4.0

AGS_IS_SFZ_SYNTH_GENERATOR()

```
#define AGS_IS_SFZ_SYNTH_GENERATOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SFZ_SYNTH_GENERATOR))
```

AGS_IS_SFZ_SYNTH_GENERATOR_CLASS()

```
#define AGS_IS_SFZ_SYNTH_GENERATOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SFZ_SYNTH_GENERATOR))
```

AGS_SFZ_SYNTH_GENERATOR()

```
#define AGS_SFZ_SYNTH_GENERATOR(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_SYNTH_GENERATOR, AgsSFZSynthGenerator))
```

AGS_SFZ_SYNTH_GENERATOR_CLASS()

```
#define AGS_SFZ_SYNTH_GENERATOR_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_SYNTH_GENERATOR, AgsSFZSynthGeneratorClass))
```

AGS_SFZ_SYNTH_GENERATOR_GET_CLASS()

```
#define AGS_SFZ_SYNTH_GENERATOR_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SFZ_SYNTH_GENERATOR, AgsSFZSynthGeneratorClass))
```

ags_sfz_synth_generator_get_type ()

```
GType
ags_sfz_synth_generator_get_type ();
```

Types and Values

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_SAMPLERATE

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_SAMPLERATE ((guint)  ↩
    AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE (AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE)
```

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_FORMAT

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_FORMAT (AGS_SOUNDCARD_DEFAULT_FORMAT)
```

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_VOLUME

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_VOLUME (1.0)
```

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE "ags-fluid-4th-order"
```

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BASE_KEY

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BASE_KEY (-48.0)
```

AGS_SFZ_SYNTH_GENERATOR_DEFAULT_TUNING

```
#define AGS_SFZ_SYNTH_GENERATOR_DEFAULT_TUNING (0.0)
```

AGS_TYPE_SFZ_SYNTH_GENERATOR

```
#define AGS_TYPE_SFZ_SYNTH_GENERATOR                (ags_sfz_synth_generator_get_type())
```

struct AgsSFZSynthGenerator

```
struct AgsSFZSynthGenerator;
```

struct AgsSFZSynthGeneratorClass

```
struct AgsSFZSynthGeneratorClass {
  AgsFunctionClass function;
};
```

Property Details

The "attack" property

```
"attack"                    guint
```

The attack to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.11

The "base-key" property

```
"base-key"                  double
```

The base key to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Allowed values: [-65535,65535]

Default value: -48

Since: 3.4.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 1024

Since: 3.4.0

The "delay" property

```
"delay"                     double
```

The delay to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Allowed values: [0,65535]

Default value: 0

Since: 3.4.11

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: NULL

Since: 3.4.0

The "format" property

```
"format"                    guint
```

The format to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 16

Since: 3.4.0

The "frame-count" property

```
"frame-count"               guint
```

The frame count to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "loop-end" property

```
"loop-end"                  guint
```

The loop end to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "loop-start" property

```
"loop-start"                guint
```

The loop start to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "pitch-type" property

```
"pitch-type"                char~*
```

The pitch type to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: "ags-fluid-4th-order"

Since: 3.9.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Default value: 48000

Since: 3.4.0

The "tuning" property

```
"tuning"                    double
```

The tuning to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Allowed values: [0,1200]

Default value: 0

Since: 3.4.0

The "volume" property

```
"volume"                    double
```

The volume to be used.

Owner: AgsSFZSynthGenerator

Flags: Read / Write

Allowed values: [0,2]

Default value: 1

Since: 3.4.11

Chapter 6

Utility functions from Fluidsynth

The utility functions from Fluidsynth.

6.1 AgsFluidUtil

AgsFluidUtil — fluid util

Stability Level

Stable, unless otherwise indicated

Functions

#define	ags_fluid_phase_set()
#define	ags_fluid_phase_set_int()
#define	ags_fluid_phase_set_float()
#define	ags_fluid_phase_from_index_fract()
#define	ags_fluid_phase_index()
#define	ags_fluid_phase_fract()
#define	ags_fluid_phase_index_round()
#define	ags_fluid_phase_fract_to_tablerow()
#define	ags_fluid_phase_double()
#define	ags_fluid_phase_incr()
#define	ags_fluid_phase_decr()
#define	ags_fluid_phase_sub_int()
#define	ags_fluid_phase_index_plusplus()
gdouble	ags_fluid_ct2hz_real ()
gdouble	ags_fluid_ct2hz ()
GType	ags_fluid_util_get_type ()

Types and Values

#define	AGS_FLUID_INTERP_BITS
#define	AGS_FLUID_INTERP_BITS_MASK
#define	AGS_FLUID_INTERP_BITS_SHIFT
#define	AGS_FLUID_FRACT_MAX

#define	AGS_FLUID_INTERP_MAX
#define	AGS_FLUID_SINC_INTERP_ORDER
#define	AGS_FLUID_CENTS_HZ_SIZE
#define	AGS_TYPE_FLUID_UTIL
struct	AgsFluidUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFluidUtil
```

Includes

```
#include <ags/audio/ags_fluid_util.h>
```

Description

Fluid utility functions.

Functions

ags_fluid_phase_set()

```
#define ags_fluid_phase_set(a,b) a=b;
```

ags_fluid_phase_set_int()

```
#define ags_fluid_phase_set_int(a, b)     ((a) = ((guint64)(b)) << 32)
```

ags_fluid_phase_set_float()

```
#define             ags_fluid_phase_set_float(a, b)
```

ags_fluid_phase_from_index_fract()

```
#define             ags_fluid_phase_from_index_fract(index, fract)
```

ags_fluid_phase_index()

```
#define             ags_fluid_phase_index(_x)
```

ags_fluid_phase_fract()

```
#define             ags_fluid_phase_fract(_x)
```

ags_fluid_phase_index_round()

```
#define             ags_fluid_phase_index_round(_x)
```

ags_fluid_phase_fract_to_tablerow()

```
#define             ags_fluid_phase_fract_to_tablerow(_x)
```

ags_fluid_phase_double()

```
#define             ags_fluid_phase_double(_x)
```

ags_fluid_phase_incr()

```
#define ags_fluid_phase_incr(a, b)  a += b
```

ags_fluid_phase_decr()

```
#define ags_fluid_phase_decr(a, b)  a -= b
```

ags_fluid_phase_sub_int()

```
#define ags_fluid_phase_sub_int(a, b)  ((a) -= (guint64)(b) << 32)
```

ags_fluid_phase_index_plusplus()

```
#define ags_fluid_phase_index_plusplus(a)  (((a) += 0x100000000LL)
```

ags_fluid_ct2hz_real ()

```
gdouble
ags_fluid_ct2hz_real (gdouble cents);
```

ags_fluid_ct2hz ()

```
gdouble
ags_fluid_ct2hz (gdouble cents);
```

ags_fluid_util_get_type ()

```
GType
ags_fluid_util_get_type (void);
```

Types and Values

AGS_FLUID_INTERP_BITS

```
#define AGS_FLUID_INTERP_BITS        (8)
```

AGS_FLUID_INTERP_BITS_MASK

```
#define AGS_FLUID_INTERP_BITS_MASK   (0xff000000)
```

AGS_FLUID_INTERP_BITS_SHIFT

```
#define AGS_FLUID_INTERP_BITS_SHIFT  (24)
```

AGS_FLUID_FRACT_MAX

```
#define AGS_FLUID_FRACT_MAX ((gdouble) 4294967296.0)
```

AGS_FLUID_INTERP_MAX

```
#define AGS_FLUID_INTERP_MAX         (2048)
```

AGS_FLUID_SINC_INTERP_ORDER

```
#define AGS_FLUID_SINC_INTERP_ORDER 7     /* 7th order constant */
```

AGS_FLUID_CENTS_HZ_SIZE

```
#define AGS_FLUID_CENTS_HZ_SIZE     (1200)
```

AGS_TYPE_FLUID_UTIL

```
#define AGS_TYPE_FLUID_UTIL         (ags_fluid_util_get_type())
```

struct AgsFluidUtil

```
struct AgsFluidUtil {
  //empty
};
```

6.2 AgsFluidInterpolateNoneUtil

AgsFluidInterpolateNoneUtil — util functions to fluid interpolate none

Stability Level

Stable, unless otherwise indicated

Functions

AgsFluidInterpolateNoneUtil *	ags_fluid_interpolate_none_util_alloc ()
gpointer	ags_fluid_interpolate_none_util_copy ()
void	ags_fluid_interpolate_none_util_free ()
gpointer	ags_fluid_interpolate_none_util_get_source ()
void	ags_fluid_interpolate_none_util_set_source ()
guint	ags_fluid_interpolate_none_util_get_source_stride ()
void	ags_fluid_interpolate_none_util_set_source_stride ()
gpointer	ags_fluid_interpolate_none_util_get_destination ()
void	ags_fluid_interpolate_none_util_set_destination ()
guint	ags_fluid_interpolate_none_util_get_destination_stride ()
void	ags_fluid_interpolate_none_util_set_destination_stride ()
guint	ags_fluid_interpolate_none_util_get_buffer_length ()
void	ags_fluid_interpolate_none_util_set_buffer_length ()
guint	ags_fluid_interpolate_none_util_get_format ()
void	ags_fluid_interpolate_none_util_set_format ()
guint	ags_fluid_interpolate_none_util_get_samplerate ()
void	ags_fluid_interpolate_none_util_set_samplerate ()
gdouble	ags_fluid_interpolate_none_util_get_base_key ()
void	ags_fluid_interpolate_none_util_set_base_key ()
gdouble	ags_fluid_interpolate_none_util_get_tuning ()
void	ags_fluid_interpolate_none_util_set_tuning ()
gdouble	ags_fluid_interpolate_none_util_get_phase_increment ()
void	ags_fluid_interpolate_none_util_set_phase_increment ()
void	ags_fluid_interpolate_none_util_pitch_s8 ()
void	ags_fluid_interpolate_none_util_pitch_s16 ()
void	ags_fluid_interpolate_none_util_pitch_s24 ()
void	ags_fluid_interpolate_none_util_pitch_s32 ()
void	ags_fluid_interpolate_none_util_pitch_s64 ()
void	ags_fluid_interpolate_none_util_pitch_float ()
void	ags_fluid_interpolate_none_util_pitch_double ()
void	ags_fluid_interpolate_none_util_pitch_complex ()
void	ags_fluid_interpolate_none_util_pitch ()
#define	AGS_FLUID_INTERPOLATE_NONE_UTIL()
GType	ags_fluid_interpolate_none_util_get_type ()

Types and Values

#define	AGS_TYPE_FLUID_INTERPOLATE_NONE_UTIL
struct	AgsFluidInterpolateNoneUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsFluidInterpolateNoneUtil
```

Includes

```
#include <ags/audio/ags_fluid_interpolate_none_util.h>
```

Description

These utility functions allow you to fill fluid interpolated none data.

Functions

ags_fluid_interpolate_none_util_alloc ()

```
AgsFluidInterpolateNoneUtil~*
ags_fluid_interpolate_none_util_alloc ();
```

Allocate AgsFluidInterpolateNoneUtil.

Returns

the newly allocated AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_copy ()

```
gpointer
ags_fluid_interpolate_none_util_copy (AgsFluidInterpolateNoneUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFluidInterpolateNoneUtil

Returns

a pointer of the new AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_free ()

```
void
ags_fluid_interpolate_none_util_free (AgsFluidInterpolateNoneUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFluidInterpolateNoneUtil	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_source ()

```
gpointer
ags_fluid_interpolate_none_util_get_source
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get source buffer of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the source buffer

Since: 3.9.6

ags_fluid_interpolate_none_util_set_source ()

```
void
ags_fluid_interpolate_none_util_set_source
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 gpointer source);
```

Set *source* buffer of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
source	the source buffer	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_source_stride ()

```
guint
ags_fluid_interpolate_none_util_get_source_stride
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get source stride of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the source buffer stride

Since: 3.9.6

ags_fluid_interpolate_none_util_set_source_stride ()

```
void
ags_fluid_interpolate_none_util_set_source_stride
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 guint source_stride);
```

Set *source* stride of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_destination ()

```
gpointer
ags_fluid_interpolate_none_util_get_destination
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get destination buffer of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the destination buffer

Since: 3.9.6

ags_fluid_interpolate_none_util_set_destination ()

```
void
ags_fluid_interpolate_none_util_set_destination
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 gpointer destination);
```

Set *destination* buffer of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
destination	the destination buffer	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_destination_stride ()

```
guint
ags_fluid_interpolate_none_util_get_destination_stride
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get destination stride of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the destination buffer stride

Since: 3.9.6

ags_fluid_interpolate_none_util_set_destination_stride ()

```
void
ags_fluid_interpolate_none_util_set_destination_stride
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 guint destination_stride);
```

Set *destination* stride of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
destination_stride	the destination buffer stride	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_buffer_length ()

```
guint
ags_fluid_interpolate_none_util_get_buffer_length
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get buffer length of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the buffer length

Since: 3.9.6

ags_fluid_interpolate_none_util_set_buffer_length ()

```
void
ags_fluid_interpolate_none_util_set_buffer_length
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 guint buffer_length);
```

Set *buffer_length* of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_format ()

```
guint
ags_fluid_interpolate_none_util_get_format
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get format of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the format

Since: 3.9.6

ags_fluid_interpolate_none_util_set_format ()

```
void
ags_fluid_interpolate_none_util_set_format
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 guint format);
```

Set *format* of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
format	the format	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_samplerate ()

```
guint
ags_fluid_interpolate_none_util_get_samplerate
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get samplerate of *fluid_interpolate_none_util*.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the samplerate

Since: 3.9.6

ags_fluid_interpolate_none_util_set_samplerate ()

```
void
ags_fluid_interpolate_none_util_set_samplerate
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 guint samplerate);
```

Set *samplerate* of *fluid_interpolate_none_util*.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_fluid_interpolate_none_util_get_base_key ()

```
gdouble
ags_fluid_interpolate_none_util_get_base_key
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get base key of *fluid_interpolate_none_util*.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the base key

Since: 4.0.0

ags_fluid_interpolate_none_util_set_base_key ()

```
void
ags_fluid_interpolate_none_util_set_base_key
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 gdouble base_key);
```

Set *base_key* of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
base_key	the base key	

Since: 4.0.0

ags_fluid_interpolate_none_util_get_tuning ()

```
gdouble
ags_fluid_interpolate_none_util_get_tuning
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get tuning of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the tuning

Since: 4.0.0

ags_fluid_interpolate_none_util_set_tuning ()

```
void
ags_fluid_interpolate_none_util_set_tuning
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 gdouble tuning);
```

Set *tuning* of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
tuning	the tuning	

Since: 4.0.0

ags_fluid_interpolate_none_util_get_phase_increment ()

```
gdouble
ags_fluid_interpolate_none_util_get_phase_increment
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Get phase increment of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	

Returns

the phase increment

Since: 3.9.6

ags_fluid_interpolate_none_util_set_phase_increment ()

```
void
ags_fluid_interpolate_none_util_set_phase_increment
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util,
                                 gdouble phase_increment);
```

Set *phase_increment* of *fluid_interpolate_none_util* .

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil	
phase_increment	the phase increment	

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_s8 ()

```
void
ags_fluid_interpolate_none_util_pitch_s8
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of signed 8 bit data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_s16 ()

```
void
ags_fluid_interpolate_none_util_pitch_s16
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of signed 16 bit data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_s24 ()

```
void
ags_fluid_interpolate_none_util_pitch_s24
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of signed 24 bit data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_s32 ()

```
void
ags_fluid_interpolate_none_util_pitch_s32
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of signed 32 bit data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_s64 ()

```
void
ags_fluid_interpolate_none_util_pitch_s64
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of signed 64 bit data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_float ()

```
void
ags_fluid_interpolate_none_util_pitch_float
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of floating point data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_double ()

```
void
ags_fluid_interpolate_none_util_pitch_double
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of double precision floating point data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch_complex ()

```
void
ags_fluid_interpolate_none_util_pitch_complex
                                (AgsFluidInterpolateNoneUtil *fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util* of complex data.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

ags_fluid_interpolate_none_util_pitch ()

```
void
ags_fluid_interpolate_none_util_pitch (AgsFluidInterpolateNoneUtil * ↩
    fluid_interpolate_none_util);
```

Pitch *fluid_interpolate_none_util*.

Parameters

fluid_interpolate_none_util	the AgsFluidInterpolateNoneUtil

Since: 3.9.6

AGS_FLUID_INTERPOLATE_NONE_UTIL()

```
#define AGS_FLUID_INTERPOLATE_NONE_UTIL(ptr) ((AgsFluidInterpolateNoneUtil *)(ptr))
```

ags_fluid_interpolate_none_util_get_type ()

```
GType
ags_fluid_interpolate_none_util_get_type
                                (void);
```

Types and Values

AGS_TYPE_FLUID_INTERPOLATE_NONE_UTIL

```
#define AGS_TYPE_FLUID_INTERPOLATE_NONE_UTIL         ( ↩
    ags_fluid_interpolate_none_util_get_type())
```

struct AgsFluidInterpolateNoneUtil

```
struct AgsFluidInterpolateNoneUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
```

```
  guint samplerate;

  gdouble base_key;
  gdouble tuning;

  gdouble phase_increment;
};
```

6.3 AgsFluidInterpolateLinearUtil

AgsFluidInterpolateLinearUtil — util functions to fluid interpolate linear

Stability Level

Stable, unless otherwise indicated

Functions

AgsFluidInterpolateLinearUtil *	ags_fluid_interpolate_linear_util_alloc ()
gpointer	ags_fluid_interpolate_linear_util_copy ()
void	ags_fluid_interpolate_linear_util_free ()
gpointer	ags_fluid_interpolate_linear_util_get_source ()
void	ags_fluid_interpolate_linear_util_set_source ()
guint	ags_fluid_interpolate_linear_util_get_source_stride ()
void	ags_fluid_interpolate_linear_util_set_source_stride ()
gpointer	ags_fluid_interpolate_linear_util_get_destination ()
void	ags_fluid_interpolate_linear_util_set_destination ()
guint	ags_fluid_interpolate_linear_util_get_destination_stride ()
void	ags_fluid_interpolate_linear_util_set_destination_stride ()
guint	ags_fluid_interpolate_linear_util_get_buffer_length ()
void	ags_fluid_interpolate_linear_util_set_buffer_length ()
guint	ags_fluid_interpolate_linear_util_get_format ()
void	ags_fluid_interpolate_linear_util_set_format ()
guint	ags_fluid_interpolate_linear_util_get_samplerate ()
void	ags_fluid_interpolate_linear_util_set_samplerate ()
gdouble	ags_fluid_interpolate_linear_util_get_base_key ()
void	ags_fluid_interpolate_linear_util_set_base_key ()
gdouble	ags_fluid_interpolate_linear_util_get_tuning ()
void	ags_fluid_interpolate_linear_util_set_tuning ()
gdouble	ags_fluid_interpolate_linear_util_get_phase_increment ()
void	ags_fluid_interpolate_linear_util_set_phase_increment ()
void	ags_fluid_interpolate_linear_util_pitch_s8 ()
void	ags_fluid_interpolate_linear_util_pitch_s16 ()
void	ags_fluid_interpolate_linear_util_pitch_s24 ()
void	ags_fluid_interpolate_linear_util_pitch_s32 ()
void	ags_fluid_interpolate_linear_util_pitch_s64 ()
void	ags_fluid_interpolate_linear_util_pitch_float ()
void	ags_fluid_interpolate_linear_util_pitch_double ()
void	ags_fluid_interpolate_linear_util_pitch_complex ()
void	ags_fluid_interpolate_linear_util_pitch ()
#define	AGS_FLUID_INTERPOLATE_LINEAR_UTIL()
GType	ags_fluid_interpolate_linear_util_get_type ()

Types and Values

#define	AGS_TYPE_FLUID_INTERPOLATE_LINEAR_UTIL
struct	AgsFluidInterpolateLinearUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFluidInterpolateLinearUtil
```

Includes

```
#include <ags/audio/ags_fluid_interpolate_linear_util.h>
```

Description

These utility functions allow you to fill fluid interpolated linear data.

Functions

ags_fluid_interpolate_linear_util_alloc ()

```
AgsFluidInterpolateLinearUtil~*
ags_fluid_interpolate_linear_util_alloc
                                ();
```

Allocate AgsFluidInterpolateLinearUtil.

Returns

the newly allocated AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_copy ()

```
gpointer
ags_fluid_interpolate_linear_util_copy
                                (AgsFluidInterpolateLinearUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFluidInterpolateLinearUtil

Returns

a pointer of the new AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_free ()

```
void
ags_fluid_interpolate_linear_util_free
                                (AgsFluidInterpolateLinearUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_source ()

```
gpointer
ags_fluid_interpolate_linear_util_get_source
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get source buffer of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Returns

the source buffer

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_source ()

```
void
ags_fluid_interpolate_linear_util_set_source
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 gpointer source);
```

Set *source* buffer of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil
source	the source buffer

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_source_stride ()

```
guint
ags_fluid_interpolate_linear_util_get_source_stride
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get source stride of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Returns

the source buffer stride

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_source_stride ()

```
void
ags_fluid_interpolate_linear_util_set_source_stride
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 guint source_stride);
```

Set *source* stride of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil
source_stride	the source buffer stride

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_destination ()

```
gpointer
ags_fluid_interpolate_linear_util_get_destination
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get destination buffer of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Returns

the destination buffer

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_destination ()

```
void
ags_fluid_interpolate_linear_util_set_destination
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 gpointer destination);
```

Set *destination* buffer of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	
destination	the destination buffer	

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_destination_stride ()

```
guint
ags_fluid_interpolate_linear_util_get_destination_stride
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get destination stride of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	

Returns

the destination buffer stride

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_destination_stride ()

```
void
ags_fluid_interpolate_linear_util_set_destination_stride
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 guint destination_stride);
```

Set *destination* stride of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	
destination_stride	the destination buffer stride	

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_buffer_length ()

```
guint
ags_fluid_interpolate_linear_util_get_buffer_length
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get buffer length of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	

Returns

the buffer length

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_buffer_length ()

```
void
ags_fluid_interpolate_linear_util_set_buffer_length
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 guint buffer_length);
```

Set *buffer_length* of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_format ()

```
guint
ags_fluid_interpolate_linear_util_get_format
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get format of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Returns

the format

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_format ()

```
void
ags_fluid_interpolate_linear_util_set_format
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 guint format);
```

Set *format* of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil
format	the format

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_samplerate ()

```
guint
ags_fluid_interpolate_linear_util_get_samplerate
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get samplerate of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Returns

the samplerate

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_samplerate ()

```
void
ags_fluid_interpolate_linear_util_set_samplerate
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 guint samplerate);
```

Set *samplerate* of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_fluid_interpolate_linear_util_get_base_key ()

```
gdouble
ags_fluid_interpolate_linear_util_get_base_key
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get base key of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	

Returns

the base key

Since: 4.0.0

ags_fluid_interpolate_linear_util_set_base_key ()

```
void
ags_fluid_interpolate_linear_util_set_base_key
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 gdouble base_key);
```

Set *base_key* of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	
base_key	the base key	

Since: 4.0.0

ags_fluid_interpolate_linear_util_get_tuning ()

```
gdouble
ags_fluid_interpolate_linear_util_get_tuning
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get tuning of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	

Returns

the tuning

Since: 4.0.0

ags_fluid_interpolate_linear_util_set_tuning ()

```
void
ags_fluid_interpolate_linear_util_set_tuning
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 gdouble tuning);
```

Set *tuning* of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil	
tuning	the tuning	

Since: 4.0.0

ags_fluid_interpolate_linear_util_get_phase_increment ()

```
gdouble
ags_fluid_interpolate_linear_util_get_phase_increment
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Get phase increment of *fluid_interpolate_linear_util* .

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Returns

the phase increment

Since: 3.9.6

ags_fluid_interpolate_linear_util_set_phase_increment ()

```
void
ags_fluid_interpolate_linear_util_set_phase_increment
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util,
                                 gdouble phase_increment);
```

Set *phase_increment* of *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil
phase_increment	the phase increment

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_s8 ()

```
void
ags_fluid_interpolate_linear_util_pitch_s8
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of signed 8 bit data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_s16 ()

```
void
ags_fluid_interpolate_linear_util_pitch_s16
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of signed 16 bit data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_s24 ()

```
void
ags_fluid_interpolate_linear_util_pitch_s24
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of signed 24 bit data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_s32 ()

```
void
ags_fluid_interpolate_linear_util_pitch_s32
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of signed 32 bit data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_s64 ()

```
void
ags_fluid_interpolate_linear_util_pitch_s64
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of signed 64 bit data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_float ()

```
void
ags_fluid_interpolate_linear_util_pitch_float
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of floating point data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_double ()

```
void
ags_fluid_interpolate_linear_util_pitch_double
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of double precision floating point data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch_complex ()

```
void
ags_fluid_interpolate_linear_util_pitch_complex
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util* of complex data.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

ags_fluid_interpolate_linear_util_pitch ()

```
void
ags_fluid_interpolate_linear_util_pitch
                                (AgsFluidInterpolateLinearUtil * ↩
                                    fluid_interpolate_linear_util);
```

Pitch *fluid_interpolate_linear_util*.

Parameters

fluid_interpolate_linear_util	the AgsFluidInterpolateLinearUtil

Since: 3.9.6

AGS_FLUID_INTERPOLATE_LINEAR_UTIL()

```
#define AGS_FLUID_INTERPOLATE_LINEAR_UTIL(ptr) ((AgsFluidInterpolateLinearUtil *)(ptr))
```

ags_fluid_interpolate_linear_util_get_type ()

```
GType
ags_fluid_interpolate_linear_util_get_type
                                (void);
```

Types and Values

AGS_TYPE_FLUID_INTERPOLATE_LINEAR_UTIL

```
#define AGS_TYPE_FLUID_INTERPOLATE_LINEAR_UTIL         ( ↩
    ags_fluid_interpolate_linear_util_get_type())
```

struct AgsFluidInterpolateLinearUtil

```
struct AgsFluidInterpolateLinearUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
```

```
  guint format;
  guint samplerate;

  gdouble base_key;
  gdouble tuning;

  gdouble phase_increment;
};
```

6.4 AgsFluidInterpolate4thOrderUtil

AgsFluidInterpolate4thOrderUtil — util functions to fluid interpolate 4th order

Stability Level

Stable, unless otherwise indicated

Functions

AgsFluidInterpolate4thOrderUtil *	ags_fluid_interpolate_4th_order_util_alloc ()
gpointer	ags_fluid_interpolate_4th_order_util_copy ()
void	ags_fluid_interpolate_4th_order_util_free ()
gpointer	ags_fluid_interpolate_4th_order_util_get_source ()
void	ags_fluid_interpolate_4th_order_util_set_source ()
guint	ags_fluid_interpolate_4th_order_util_get_source_stride ()
void	ags_fluid_interpolate_4th_order_util_set_source_stride ()
gpointer	ags_fluid_interpolate_4th_order_util_get_destination ()
void	ags_fluid_interpolate_4th_order_util_set_destination ()
guint	ags_fluid_interpolate_4th_order_util_get_destination_stride ()
void	ags_fluid_interpolate_4th_order_util_set_destination_stride ()
guint	ags_fluid_interpolate_4th_order_util_get_buffer_length ()
void	ags_fluid_interpolate_4th_order_util_set_buffer_length ()
guint	ags_fluid_interpolate_4th_order_util_get_format ()
void	ags_fluid_interpolate_4th_order_util_set_format ()
guint	ags_fluid_interpolate_4th_order_util_get_samplerate ()
void	ags_fluid_interpolate_4th_order_util_set_samplerate ()
gdouble	ags_fluid_interpolate_4th_order_util_get_base_key ()
void	ags_fluid_interpolate_4th_order_util_set_base_key ()
gdouble	ags_fluid_interpolate_4th_order_util_get_tuning ()
void	ags_fluid_interpolate_4th_order_util_set_tuning ()
gdouble	ags_fluid_interpolate_4th_order_util_get_phase_increment ()
void	ags_fluid_interpolate_4th_order_util_set_phase_increment ()
void	ags_fluid_interpolate_4th_order_util_pitch_s8 ()
void	ags_fluid_interpolate_4th_order_util_pitch_s16 ()
void	ags_fluid_interpolate_4th_order_util_pitch_s24 ()
void	ags_fluid_interpolate_4th_order_util_pitch_s32 ()
void	ags_fluid_interpolate_4th_order_util_pitch_s64 ()
void	ags_fluid_interpolate_4th_order_util_pitch_float ()
void	ags_fluid_interpolate_4th_order_util_pitch_double ()
void	ags_fluid_interpolate_4th_order_util_pitch_complex ()
void	ags_fluid_interpolate_4th_order_util_pitch ()
#define	AGS_FLUID_INTERPOLATE_4TH_ORDER_UTIL()

GType	ags_fluid_interpolate_4th_order_util_get_type ()

Types and Values

#define	AGS_TYPE_FLUID_INTERPOLATE_4TH_ORDER_UTIL
struct	AgsFluidInterpolate4thOrderUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFluidInterpolate4thOrderUtil
```

Includes

```
#include <ags/audio/ags_fluid_interpolate_4th_order_util.h>
```

Description

These utility functions allow you to fill fluid interpolated 4th order data.

Functions

ags_fluid_interpolate_4th_order_util_alloc ()

```
AgsFluidInterpolate4thOrderUtil~*
ags_fluid_interpolate_4th_order_util_alloc
                                ();
```

Allocate AgsFluidInterpolate4thOrderUtil.

Returns

the newly allocated AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_copy ()

```
gpointer
ags_fluid_interpolate_4th_order_util_copy
                                (AgsFluidInterpolate4thOrderUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFluidInterpolate4thOrderUtil

Returns

a pointer of the new AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_free ()

```
void
ags_fluid_interpolate_4th_order_util_free
                                (AgsFluidInterpolate4thOrderUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_source ()

```
gpointer
ags_fluid_interpolate_4th_order_util_get_source
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get source buffer of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Returns

the source buffer

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_source ()

```
void
ags_fluid_interpolate_4th_order_util_set_source
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 gpointer source);
```

Set *source* buffer of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
source	the source buffer	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_source_stride ()

```
guint
ags_fluid_interpolate_4th_order_util_get_source_stride
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get source stride of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the source buffer stride

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_source_stride ()

```
void
ags_fluid_interpolate_4th_order_util_set_source_stride
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 guint source_stride);
```

Set *source* stride of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_destination ()

```
gpointer
ags_fluid_interpolate_4th_order_util_get_destination
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get destination buffer of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the destination buffer

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_destination ()

```
void
ags_fluid_interpolate_4th_order_util_set_destination
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 gpointer destination);
```

Set *destination* buffer of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
destination	the destination buffer	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_destination_stride ()

```
guint
ags_fluid_interpolate_4th_order_util_get_destination_stride
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get destination stride of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the destination buffer stride

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_destination_stride ()

```
void
ags_fluid_interpolate_4th_order_util_set_destination_stride
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 guint destination_stride);
```

Set *destination* stride of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
destination_stride	the destination buffer stride	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_buffer_length ()

```
guint
ags_fluid_interpolate_4th_order_util_get_buffer_length
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get buffer length of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the buffer length

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_buffer_length ()

```
void
ags_fluid_interpolate_4th_order_util_set_buffer_length
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 guint buffer_length);
```

Set *buffer_length* of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_format ()

```
guint
ags_fluid_interpolate_4th_order_util_get_format
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get format of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the format

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_format ()

```
void
ags_fluid_interpolate_4th_order_util_set_format
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 guint format);
```

Set *format* of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
format	the format	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_samplerate ()

```
guint
ags_fluid_interpolate_4th_order_util_get_samplerate
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get samplerate of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the samplerate

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_samplerate ()

```
void
ags_fluid_interpolate_4th_order_util_set_samplerate
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 guint samplerate);
```

Set *samplerate* of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_get_base_key ()

```
gdouble
ags_fluid_interpolate_4th_order_util_get_base_key
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get base key of *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the base key

Since: 4.0.0

ags_fluid_interpolate_4th_order_util_set_base_key ()

```
void
ags_fluid_interpolate_4th_order_util_set_base_key
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 gdouble base_key);
```

Set *base_key* of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
base_key	the base key	

Since: 4.0.0

ags_fluid_interpolate_4th_order_util_get_tuning ()

```
gdouble
ags_fluid_interpolate_4th_order_util_get_tuning
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get tuning of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the tuning

Since: 4.0.0

ags_fluid_interpolate_4th_order_util_set_tuning ()

```
void
ags_fluid_interpolate_4th_order_util_set_tuning
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 gdouble tuning);
```

Set *tuning* of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
tuning	the tuning	

Since: 4.0.0

ags_fluid_interpolate_4th_order_util_get_phase_increment ()

```
gdouble
ags_fluid_interpolate_4th_order_util_get_phase_increment
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Get phase increment of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	

Returns

the phase increment

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_set_phase_increment ()

```
void
ags_fluid_interpolate_4th_order_util_set_phase_increment
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util,
                                 gdouble phase_increment);
```

Set *phase_increment* of *fluid_interpolate_4th_order_util* .

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil	
phase_increment	the phase increment	

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_s8 ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_s8
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of signed 8 bit data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_s16 ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_s16
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of signed 16 bit data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_s24 ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_s24
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of signed 24 bit data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_s32 ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_s32
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of signed 32 bit data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_s64 ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_s64
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of signed 64 bit data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_float ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_float
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of floating point data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_double ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_double
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of double precision floating point data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch_complex ()

```
void
ags_fluid_interpolate_4th_order_util_pitch_complex
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util* of complex data.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_4th_order_util_pitch ()

```
void
ags_fluid_interpolate_4th_order_util_pitch
                                (AgsFluidInterpolate4thOrderUtil * ↩
                                    fluid_interpolate_4th_order_util);
```

Pitch *fluid_interpolate_4th_order_util*.

Parameters

fluid_interpolate_4th_order_util	the AgsFluidInterpolate4thOrderUtil

Since: 3.9.6

AGS_FLUID_INTERPOLATE_4TH_ORDER_UTIL()

```
#define AGS_FLUID_INTERPOLATE_4TH_ORDER_UTIL(ptr) ((AgsFluidInterpolate4thOrderUtil *)(ptr) ↩
    )
```

ags_fluid_interpolate_4th_order_util_get_type ()

```
GType
ags_fluid_interpolate_4th_order_util_get_type
                                (void);
```

Types and Values

AGS_TYPE_FLUID_INTERPOLATE_4TH_ORDER_UTIL

```
#define AGS_TYPE_FLUID_INTERPOLATE_4TH_ORDER_UTIL         ( ↩
    ags_fluid_interpolate_4th_order_util_get_type())
```

struct AgsFluidInterpolate4thOrderUtil

```
struct AgsFluidInterpolate4thOrderUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble base_key;
  gdouble tuning;

  gdouble phase_increment;
};
```

6.5 AgsFluidInterpolate7thOrderUtil

AgsFluidInterpolate7thOrderUtil — util functions to fluid interpolate 7th order

Stability Level

Stable, unless otherwise indicated

Functions

AgsFluidInterpolate7thOrderUtil *	ags_fluid_interpolate_7th_order_util_alloc ()
gpointer	ags_fluid_interpolate_7th_order_util_copy ()
void	ags_fluid_interpolate_7th_order_util_free ()
gpointer	ags_fluid_interpolate_7th_order_util_get_source ()
void	ags_fluid_interpolate_7th_order_util_set_source ()
guint	ags_fluid_interpolate_7th_order_util_get_source_stride ()
void	ags_fluid_interpolate_7th_order_util_set_source_stride ()
gpointer	ags_fluid_interpolate_7th_order_util_get_destination ()
void	ags_fluid_interpolate_7th_order_util_set_destination ()
guint	ags_fluid_interpolate_7th_order_util_get_destination_stride ()
void	ags_fluid_interpolate_7th_order_util_set_destination_stride ()
guint	ags_fluid_interpolate_7th_order_util_get_buffer_length ()
void	ags_fluid_interpolate_7th_order_util_set_buffer_length ()
guint	ags_fluid_interpolate_7th_order_util_get_format ()
void	ags_fluid_interpolate_7th_order_util_set_format ()
guint	ags_fluid_interpolate_7th_order_util_get_samplerate ()

void	ags_fluid_interpolate_7th_order_util_set_samplerate ()
gdouble	ags_fluid_interpolate_7th_order_util_get_base_key ()
void	ags_fluid_interpolate_7th_order_util_set_base_key ()
gdouble	ags_fluid_interpolate_7th_order_util_get_tuning ()
void	ags_fluid_interpolate_7th_order_util_set_tuning ()
gdouble	ags_fluid_interpolate_7th_order_util_get_phase_increment ()
void	ags_fluid_interpolate_7th_order_util_set_phase_increment ()
void	ags_fluid_interpolate_7th_order_util_pitch_s8 ()
void	ags_fluid_interpolate_7th_order_util_pitch_s16 ()
void	ags_fluid_interpolate_7th_order_util_pitch_s24 ()
void	ags_fluid_interpolate_7th_order_util_pitch_s32 ()
void	ags_fluid_interpolate_7th_order_util_pitch_s64 ()
void	ags_fluid_interpolate_7th_order_util_pitch_float ()
void	ags_fluid_interpolate_7th_order_util_pitch_double ()
void	ags_fluid_interpolate_7th_order_util_pitch_complex ()
void	ags_fluid_interpolate_7th_order_util_pitch ()
#define	AGS_FLUID_INTERPOLATE_7TH_ORDER_UTIL()
GType	ags_fluid_interpolate_7th_order_util_get_type ()

Types and Values

#define	AGS_TYPE_FLUID_INTERPOLATE_7TH_ORDER_UTIL
struct	AgsFluidInterpolate7thOrderUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsFluidInterpolate7thOrderUtil
```

Includes

```
#include <ags/audio/ags_fluid_interpolate_7th_order_util.h>
```

Description

These utility functions allow you to fill fluid interpolated 7th order data.

Functions

ags_fluid_interpolate_7th_order_util_alloc ()

```
AgsFluidInterpolate7thOrderUtil~*
ags_fluid_interpolate_7th_order_util_alloc
                                ();
```

Allocate AgsFluidInterpolate7thOrderUtil.

Returns

the newly allocated AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_copy ()

```
gpointer
ags_fluid_interpolate_7th_order_util_copy
                                (AgsFluidInterpolate7thOrderUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFluidInterpolate7thOrderUtil

Returns

a pointer of the new AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_free ()

```
void
ags_fluid_interpolate_7th_order_util_free
                                (AgsFluidInterpolate7thOrderUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_source ()

```
gpointer
ags_fluid_interpolate_7th_order_util_get_source
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get source buffer of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Returns

the source buffer

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_source ()

```
void
ags_fluid_interpolate_7th_order_util_set_source
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 gpointer source);
```

Set *source* buffer of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
source	the source buffer	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_source_stride ()

```
guint
ags_fluid_interpolate_7th_order_util_get_source_stride
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get source stride of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the source buffer stride

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_source_stride ()

```
void
ags_fluid_interpolate_7th_order_util_set_source_stride
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 guint source_stride);
```

Set *source* stride of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_destination ()

```
gpointer
ags_fluid_interpolate_7th_order_util_get_destination
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get destination buffer of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the destination buffer

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_destination ()

```
void
ags_fluid_interpolate_7th_order_util_set_destination
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 gpointer destination);
```

Set *destination* buffer of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
destination	the destination buffer	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_destination_stride ()

```
guint
ags_fluid_interpolate_7th_order_util_get_destination_stride
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get destination stride of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the destination buffer stride

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_destination_stride ()

```
void
ags_fluid_interpolate_7th_order_util_set_destination_stride
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 guint destination_stride);
```

Set *destination* stride of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
destination_stride	the destination buffer stride	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_buffer_length ()

```
guint
ags_fluid_interpolate_7th_order_util_get_buffer_length
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get buffer length of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the buffer length

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_buffer_length ()

```
void
ags_fluid_interpolate_7th_order_util_set_buffer_length
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 guint buffer_length);
```

Set *buffer_length* of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_format ()

```
guint
ags_fluid_interpolate_7th_order_util_get_format
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get format of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the format

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_format ()

```
void
ags_fluid_interpolate_7th_order_util_set_format
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 guint format);
```

Set *format* of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
format	the format	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_samplerate ()

```
guint
ags_fluid_interpolate_7th_order_util_get_samplerate
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get samplerate of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the samplerate

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_samplerate ()

```
void
ags_fluid_interpolate_7th_order_util_set_samplerate
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 guint samplerate);
```

Set *samplerate* of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_get_base_key ()

```
gdouble
ags_fluid_interpolate_7th_order_util_get_base_key
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get base key of *fluid_interpolate_7th_order_util* .

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the base key

Since: 4.0.0

ags_fluid_interpolate_7th_order_util_set_base_key ()

```
void
ags_fluid_interpolate_7th_order_util_set_base_key
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 gdouble base_key);
```

Set *base_key* of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
base_key	the base key	

Since: 4.0.0

ags_fluid_interpolate_7th_order_util_get_tuning ()

```
gdouble
ags_fluid_interpolate_7th_order_util_get_tuning
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get tuning of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the tuning

Since: 4.0.0

ags_fluid_interpolate_7th_order_util_set_tuning ()

```
void
ags_fluid_interpolate_7th_order_util_set_tuning
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 gdouble tuning);
```

Set *tuning* of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
tuning	the tuning	

Since: 4.0.0

ags_fluid_interpolate_7th_order_util_get_phase_increment ()

```
gdouble
ags_fluid_interpolate_7th_order_util_get_phase_increment
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Get phase increment of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	

Returns

the phase increment

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_set_phase_increment ()

```
void
ags_fluid_interpolate_7th_order_util_set_phase_increment
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util,
                                 gdouble phase_increment);
```

Set *phase_increment* of *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil	
phase_increment	the phase increment	

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_s8 ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_s8
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of signed 8 bit data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_s16 ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_s16
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of signed 16 bit data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_s24 ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_s24
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of signed 24 bit data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_s32 ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_s32
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of signed 32 bit data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_s64 ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_s64
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of signed 64 bit data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_float ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_float
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of floating point data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_double ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_double
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of double precision floating point data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch_complex ()

```
void
ags_fluid_interpolate_7th_order_util_pitch_complex
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util* of complex data.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

ags_fluid_interpolate_7th_order_util_pitch ()

```
void
ags_fluid_interpolate_7th_order_util_pitch
                                (AgsFluidInterpolate7thOrderUtil * ↩
                                    fluid_interpolate_7th_order_util);
```

Pitch *fluid_interpolate_7th_order_util*.

Parameters

fluid_interpolate_7th_order_util	the AgsFluidInterpolate7thOrderUtil

Since: 3.9.6

AGS_FLUID_INTERPOLATE_7TH_ORDER_UTIL()

```
#define AGS_FLUID_INTERPOLATE_7TH_ORDER_UTIL(ptr) ((AgsFluidInterpolate7thOrderUtil *)(ptr) ↩
    )
```

ags_fluid_interpolate_7th_order_util_get_type ()

```
GType
ags_fluid_interpolate_7th_order_util_get_type
                                (void);
```

Types and Values

AGS_TYPE_FLUID_INTERPOLATE_7TH_ORDER_UTIL

```
#define AGS_TYPE_FLUID_INTERPOLATE_7TH_ORDER_UTIL         ( ↩
    ags_fluid_interpolate_7th_order_util_get_type())
```

struct AgsFluidInterpolate7thOrderUtil

```
struct AgsFluidInterpolate7thOrderUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  gdouble base_key;
  gdouble tuning;

  gdouble phase_increment;
};
```

6.6 AgsFluidIIRFilterUtil

AgsFluidIIRFilterUtil — util functions to fluid IIR filter

Stability Level

Stable, unless otherwise indicated

Functions

AgsFluidIIRFilterUtil *	ags_fluid_iir_filter_util_alloc ()
gpointer	ags_fluid_iir_filter_util_copy ()
void	ags_fluid_iir_filter_util_free ()
gpointer	ags_fluid_iir_filter_util_get_source ()
void	ags_fluid_iir_filter_util_set_source ()
guint	ags_fluid_iir_filter_util_get_source_stride ()
void	ags_fluid_iir_filter_util_set_source_stride ()
gpointer	ags_fluid_iir_filter_util_get_destination ()

void	ags_fluid_iir_filter_util_set_destination ()
guint	ags_fluid_iir_filter_util_get_destination_stride ()
void	ags_fluid_iir_filter_util_set_destination_stride ()
guint	ags_fluid_iir_filter_util_get_buffer_length ()
void	ags_fluid_iir_filter_util_set_buffer_length ()
guint	ags_fluid_iir_filter_util_get_format ()
void	ags_fluid_iir_filter_util_set_format ()
guint	ags_fluid_iir_filter_util_get_samplerate ()
void	ags_fluid_iir_filter_util_set_samplerate ()
guint	ags_fluid_iir_filter_util_get_filter_type ()
void	ags_fluid_iir_filter_util_set_filter_type ()
guint	ags_fluid_iir_filter_util_get_flags ()
void	ags_fluid_iir_filter_util_set_flags ()
gboolean	ags_fluid_iir_filter_util_get_filter_startup ()
void	ags_fluid_iir_filter_util_set_filter_startup ()
gdouble	ags_fluid_iir_filter_util_get_q_lin ()
void	ags_fluid_iir_filter_util_set_q_lin ()
gdouble	ags_fluid_iir_filter_util_get_filter_gain ()
void	ags_fluid_iir_filter_util_set_filter_gain ()
void	ags_fluid_iir_filter_util_process_s8 ()
void	ags_fluid_iir_filter_util_process_s16 ()
void	ags_fluid_iir_filter_util_process_s24 ()
void	ags_fluid_iir_filter_util_process_s32 ()
void	ags_fluid_iir_filter_util_process_s64 ()
void	ags_fluid_iir_filter_util_process_float ()
void	ags_fluid_iir_filter_util_process_double ()
void	ags_fluid_iir_filter_util_process_complex ()
void	ags_fluid_iir_filter_util_process ()
#define	AGS_FLUID_IIR_FILTER_UTIL()
GType	ags_fluid_iir_filter_type_get_type ()
GType	ags_fluid_iir_filter_util_get_type ()

Types and Values

enum	AgsFluidIIRFilterType
#define	AGS_TYPE_FLUID_IIR_FILTER_UTIL
struct	AgsFluidIIRFilterUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsFluidIIRFilterUtil
GFlags
&#x2570;&#x2500;&#x2500; AgsFluidIIRFilterType
```

Includes

```
#include <ags/audio/ags_fluid_iir_filter_util.h>
```

Description

These utility functions allow you to calc and apply fluid IIR filter data.

Functions

ags_fluid_iir_filter_util_alloc ()

```
AgsFluidIIRFilterUtil~*
ags_fluid_iir_filter_util_alloc ();
```

Allocate AgsFluidIIRFilterUtil.

Returns

the newly allocated AgsFluidIIRFilterUtil

Since: 3.9.6

ags_fluid_iir_filter_util_copy ()

```
gpointer
ags_fluid_iir_filter_util_copy (AgsFluidIIRFilterUtil *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsFluidIIRFilterUtil

Returns

a pointer of the new AgsFluidIIRFilterUtil

Since: 3.9.6

ags_fluid_iir_filter_util_free ()

```
void
ags_fluid_iir_filter_util_free (AgsFluidIIRFilterUtil *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsFluidIIRFilterUtil

Since: 3.9.6

ags_fluid_iir_filter_util_get_source ()

```
gpointer
ags_fluid_iir_filter_util_get_source (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get source buffer of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Returns

the source buffer

Since: 3.9.6

ags_fluid_iir_filter_util_set_source ()

```
void
ags_fluid_iir_filter_util_set_source (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                      gpointer source);
```

Set *source* buffer of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil
source	the source buffer

Since: 3.9.6

ags_fluid_iir_filter_util_get_source_stride ()

```
guint
ags_fluid_iir_filter_util_get_source_stride
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get source stride of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Returns

the source buffer stride

Since: 3.9.6

ags_fluid_iir_filter_util_set_source_stride ()

```
void
ags_fluid_iir_filter_util_set_source_stride
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 guint source_stride);
```

Set *source* stride of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
source_stride	the source buffer stride	

Since: 3.9.6

ags_fluid_iir_filter_util_get_destination ()

```
gpointer
ags_fluid_iir_filter_util_get_destination
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get destination buffer of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the destination buffer

Since: 3.9.6

ags_fluid_iir_filter_util_set_destination ()

```
void
ags_fluid_iir_filter_util_set_destination
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 gpointer destination);
```

Set *destination* buffer of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
destination	the destination buffer	

Since: 3.9.6

ags_fluid_iir_filter_util_get_destination_stride ()

```
guint
ags_fluid_iir_filter_util_get_destination_stride
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get destination stride of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Returns

the destination buffer stride

Since: 3.9.6

ags_fluid_iir_filter_util_set_destination_stride ()

```
void
ags_fluid_iir_filter_util_set_destination_stride
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 guint destination_stride);
```

Set *destination* stride of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil
destination_stride	the destination buffer stride

Since: 3.9.6

ags_fluid_iir_filter_util_get_buffer_length ()

```
guint
ags_fluid_iir_filter_util_get_buffer_length
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get buffer length of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Returns

the buffer length

Since: 3.9.6

ags_fluid_iir_filter_util_set_buffer_length ()

```
void
ags_fluid_iir_filter_util_set_buffer_length
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 guint buffer_length);
```

Set *buffer_length* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
buffer_length	the buffer length	

Since: 3.9.6

ags_fluid_iir_filter_util_get_format ()

```
guint
ags_fluid_iir_filter_util_get_format (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get format of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the format

Since: 3.9.6

ags_fluid_iir_filter_util_set_format ()

```
void
ags_fluid_iir_filter_util_set_format (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                      guint format);
```

Set *format* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
format	the format	

Since: 3.9.6

ags_fluid_iir_filter_util_get_samplerate ()

```
guint
ags_fluid_iir_filter_util_get_samplerate
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get samplerate of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the samplerate

Since: 3.9.6

ags_fluid_iir_filter_util_set_samplerate ()

```
void
ags_fluid_iir_filter_util_set_samplerate
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 guint samplerate);
```

Set *samplerate* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
samplerate	the samplerate	

Since: 3.9.6

ags_fluid_iir_filter_util_get_filter_type ()

```
guint
ags_fluid_iir_filter_util_get_filter_type
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get filter type of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the filter type

Since: 3.9.7

ags_fluid_iir_filter_util_set_filter_type ()

```
void
ags_fluid_iir_filter_util_set_filter_type
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 guint filter_type);
```

Set *filter_type* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
filter_type	the filter type	

Since: 3.9.7

ags_fluid_iir_filter_util_get_flags ()

```
guint
ags_fluid_iir_filter_util_get_flags (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get flags of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the flags

Since: 3.9.7

ags_fluid_iir_filter_util_set_flags ()

```
void
ags_fluid_iir_filter_util_set_flags (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                     guint flags);
```

Set *flags* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
flags	the flags	

Since: 3.9.7

ags_fluid_iir_filter_util_get_filter_startup ()

```
gboolean
ags_fluid_iir_filter_util_get_filter_startup
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get filter startup of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

TRUE if the filter does startup, otherwise FALSE

Since: 3.9.7

ags_fluid_iir_filter_util_set_filter_startup ()

```
void
ags_fluid_iir_filter_util_set_filter_startup
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 gboolean filter_startup);
```

Set *filter_startup* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
filter_startup	the filter startup	

Since: 3.9.7

ags_fluid_iir_filter_util_get_q_lin ()

```
gdouble
ags_fluid_iir_filter_util_get_q_lin (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get q-lin of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the q-lin

Since: 3.9.7

ags_fluid_iir_filter_util_set_q_lin ()

```
void
ags_fluid_iir_filter_util_set_q_lin (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                     gdouble q_lin);
```

Set *q_lin* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
q_lin	the q-lin	

Since: 3.9.7

ags_fluid_iir_filter_util_get_filter_gain ()

```
gdouble
ags_fluid_iir_filter_util_get_filter_gain
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Get filter gain of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	

Returns

the filter gain

Since: 3.9.7

ags_fluid_iir_filter_util_set_filter_gain ()

```
void
ags_fluid_iir_filter_util_set_filter_gain
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util,
                                 gdouble filter_gain);
```

Set *filter_gain* of *fluid_iir_filter_util* .

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil	
filter_gain	the filter gain	

Since: 3.9.7

ags_fluid_iir_filter_util_process_s8 ()

```
void
ags_fluid_iir_filter_util_process_s8 (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of signed 8 bit data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_s16 ()

```
void
ags_fluid_iir_filter_util_process_s16 (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of signed 16 bit data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_s24 ()

```
void
ags_fluid_iir_filter_util_process_s24 (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of signed 24 bit data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_s32 ()

```
void
ags_fluid_iir_filter_util_process_s32 (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of signed 32 bit data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_s64 ()

```
void
ags_fluid_iir_filter_util_process_s64 (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of signed 64 bit data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_float ()

```
void
ags_fluid_iir_filter_util_process_float
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of floating point data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_double ()

```
void
ags_fluid_iir_filter_util_process_double
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of double precision floating point data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process_complex ()

```
void
ags_fluid_iir_filter_util_process_complex
                                (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util* of complex data.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

ags_fluid_iir_filter_util_process ()

```
void
ags_fluid_iir_filter_util_process (AgsFluidIIRFilterUtil *fluid_iir_filter_util);
```

Process *fluid_iir_filter_util*.

Parameters

fluid_iir_filter_util	the AgsFluidIIRFilterUtil

Since: 3.9.7

AGS_FLUID_IIR_FILTER_UTIL()

```
#define AGS_FLUID_IIR_FILTER_UTIL(ptr)          ((AgsFluidIIRFilterUtil *)(ptr))
```

ags_fluid_iir_filter_type_get_type ()

```
GType
ags_fluid_iir_filter_type_get_type ();
```

ags_fluid_iir_filter_util_get_type ()

```
GType
ags_fluid_iir_filter_util_get_type (void);
```

Types and Values

enum AgsFluidIIRFilterType

Enum values to specify filter type of AgsFluidIIRFilterUtil.

Members

AGS_FLUID_IIR_DISABLED	disabled
AGS_FLUID_IIR_LOWPASS	low-pass
AGS_FLUID_IIR_HIGHPASS	high-pass
AGS_FLUID_IIR_LAST	last en-try

AGS_TYPE_FLUID_IIR_FILTER_UTIL

```
#define AGS_TYPE_FLUID_IIR_FILTER_UTIL          (ags_fluid_iir_filter_util_get_type())
```

struct AgsFluidIIRFilterUtil

```
struct AgsFluidIIRFilterUtil {
  gpointer source;
  guint source_stride;

  gpointer destination;
  guint destination_stride;

  guint buffer_length;
  guint format;
  guint samplerate;

  guint filter_type;

  guint flags;

  gdouble b02;
  gdouble b1;
  gdouble a1;
  gdouble a2;
  gdouble b02_incr;
  gdouble b1_incr;
  gdouble a1_incr;
  gdouble a2_incr;

  gint filter_coeff_incr_count;
  gint compensate_incr;

  gdouble hist1;
  gdouble hist2;

  gboolean filter_startup;

  gdouble fres;
  gdouble last_fres;

  gdouble q_lin;
  gdouble filter_gain;
};
```

Chapter 7

Frequency utils

The frequency util objects.

7.1 AgsFrequencyMapManager

AgsFrequencyMapManager — frequency map manager

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FREQUENCY_MAP_MANAGER_GET_OBJ_MUTEX()
gboolean	ags_frequency_map_manager_test_flags ()
void	ags_frequency_map_manager_set_flags ()
void	ags_frequency_map_manager_unset_flags ()
void	ags_frequency_map_manager_add_frequency_map ()
void	ags_frequency_map_manager_add_factorized_frequency_map ()
AgsFrequencyMap *	ags_frequency_map_manager_find_frequency_map ()
AgsFrequencyMap *	ags_frequency_map_manager_find_factorized_frequency_map ()
void	ags_frequency_map_manager_load_default ()
AgsFrequencyMapManager *	ags_frequency_map_manager_get_instance ()
AgsFrequencyMapManager *	ags_frequency_map_manager_new ()
#define	AGS_FREQUENCY_MAP_MANAGER()
#define	AGS_FREQUENCY_MAP_MANAGER_CLASS()
#define	AGS_FREQUENCY_MAP_MANAGER_GET_CLASS()
#define	AGS_IS_FREQUENCY_MAP_MANAGER()
#define	AGS_IS_FREQUENCY_MAP_MANAGER_CLASS()
GType	ags_frequency_map_manager_get_type ()

Types and Values

#define	AGS_FREQUENCY_MAP_MANAGER_EQUINOX
enum	AgsFrequencyMapManagerFlags
#define	AGS_TYPE_FREQUENCY_MAP_MANAGER

struct	AgsFrequencyMapManager
struct	AgsFrequencyMapManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFrequencyMapManager
```

Includes

```
#include <ags/audio/ags_frequency_map_manager.h>
```

Description

AgsFrequencyMapManager is a singleton that organizes your frequency maps.

Functions

AGS_FREQUENCY_MAP_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_FREQUENCY_MAP_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsFrequencyMapManager *) obj)-> ↩
    obj_mutex))
```

ags_frequency_map_manager_test_flags ()

```
gboolean
ags_frequency_map_manager_test_flags (AgsFrequencyMapManager *frequency_map_manager,
                                      guint flags);
```

Test *flags* to be set on *recall* .

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_frequency_map_manager_set_flags ()

```
void
ags_frequency_map_manager_set_flags (AgsFrequencyMapManager *frequency_map_manager,
                                     guint flags);
```

Set flags.

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
flags	the flags	

Since: 3.0.0

ags_frequency_map_manager_unset_flags ()

```
void
ags_frequency_map_manager_unset_flags (AgsFrequencyMapManager *frequency_map_manager,
                                       guint flags);
```

Unset flags.

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
flags	the flags	

Since: 3.0.0

ags_frequency_map_manager_add_frequency_map ()

```
void
ags_frequency_map_manager_add_frequency_map
                                (AgsFrequencyMapManager *frequency_map_manager,
                                 AgsFrequencyMap *frequency_map);
```

Add *frequency_map* to *frequency_map_manager* .

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
frequency_map	the AgsFrequencyMap	

Since: 3.0.0

ags_frequency_map_manager_add_factorized_frequency_map ()

```
void
ags_frequency_map_manager_add_factorized_frequency_map
                                (AgsFrequencyMapManager *frequency_map_manager,
                                 AgsFrequencyMap *frequency_map);
```

Add *frequency_map* to *frequency_map_manager* .

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
frequency_map	the AgsFrequencyMap	

Since: 3.0.0

ags_frequency_map_manager_find_frequency_map ()

```
AgsFrequencyMap~*
ags_frequency_map_manager_find_frequency_map
                                (AgsFrequencyMapManager *frequency_map_manager,
                                 guint samplerate,
                                 guint buffer_size,
                                 gdouble freq);
```

Find AgsFrequencyMap by *samplerate* , *buffer_size* and *freq* .

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
samplerate	the samplerate to match	
buffer_size	the buffer size to match	
freq	the frequency to match	

Returns

the matching AgsFrequencyMap if found, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_frequency_map_manager_find_factorized_frequency_map ()

```
AgsFrequencyMap~*
ags_frequency_map_manager_find_factorized_frequency_map
                                (AgsFrequencyMapManager *frequency_map_manager,
                                 guint samplerate,
                                 guint buffer_size);
```

Find AgsFrequencyMap by *samplerate* and *buffer_size* .

Parameters

frequency_map_manager	the AgsFrequencyMapManager	
samplerate	the samplerate to match	
buffer_size	the buffer size to match	

Returns

the matching AgsFrequencyMap if found, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_frequency_map_manager_load_default ()

```
void
ags_frequency_map_manager_load_default
                                (AgsFrequencyMapManager *frequency_map_manager);
```

Load default AgsFrequencyMap.

Parameters

frequency_map_manager	the AgsFrequencyMapManager

Since: 3.0.0

ags_frequency_map_manager_get_instance ()

```
AgsFrequencyMapManager~*
ags_frequency_map_manager_get_instance
                                ();
```

Get instance.

Returns

the AgsFrequencyMapManager.

[transfer none]

Since: 3.0.0

ags_frequency_map_manager_new ()

```
AgsFrequencyMapManager~*
ags_frequency_map_manager_new ();
```

Create a new instance of AgsFrequencyMapManager

Returns

the new AgsFrequencyMapManager

Since: 3.0.0

AGS_FREQUENCY_MAP_MANAGER()

```
#define AGS_FREQUENCY_MAP_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FREQUENCY_MAP_MANAGER, AgsFrequencyMapManager))
```

AGS_FREQUENCY_MAP_MANAGER_CLASS()

```
#define AGS_FREQUENCY_MAP_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FREQUENCY_MAP_MANAGER, AgsFrequencyMapManagerClass))
```

AGS_FREQUENCY_MAP_MANAGER_GET_CLASS()

```
#define AGS_FREQUENCY_MAP_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FREQUENCY_MAP_MANAGER, AgsFrequencyMapManagerClass))
```

AGS_IS_FREQUENCY_MAP_MANAGER()

```
#define AGS_IS_FREQUENCY_MAP_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FREQUENCY_MAP_MANAGER))
```

AGS_IS_FREQUENCY_MAP_MANAGER_CLASS()

```
#define AGS_IS_FREQUENCY_MAP_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FREQUENCY_MAP_MANAGER))
```

ags_frequency_map_manager_get_type ()

```
GType
ags_frequency_map_manager_get_type (void);
```

Types and Values

AGS_FREQUENCY_MAP_MANAGER_EQUINOX

```
#define AGS_FREQUENCY_MAP_MANAGER_EQUINOX (22000)
```

enum AgsFrequencyMapManagerFlags

Enum values to control the behavior or indicate internal state of AgsFrequencyMapManager by enable/disable as flags.

Members

AGS_FREQUENCY_MAP_MANAGER_PRESERVE_FREQUENCY_MAP	if set preserve frequency maps, otherwise destroy

AGS_TYPE_FREQUENCY_MAP_MANAGER

```
#define AGS_TYPE_FREQUENCY_MAP_MANAGER                (ags_frequency_map_manager_get_type() ↩
    )
```

struct AgsFrequencyMapManager

```
struct AgsFrequencyMapManager;
```

struct AgsFrequencyMapManagerClass

```
struct AgsFrequencyMapManagerClass {
  GObjectClass gobject;
};
```

7.2 AgsFrequencyMap

AgsFrequencyMap — Contains the audio data for a given frequency

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FREQUENCY_MAP_GET_OBJ_MUTEX()
gint	ags_frequency_map_sort_func ()
gboolean	ags_frequency_map_test_flags ()
void	ags_frequency_map_set_flags ()
void	ags_frequency_map_unset_flags ()
void	ags_frequency_map_process ()
void	ags_frequency_map_factorize ()
void	ags_frequency_map_compute_max_likelihood ()
AgsFrequencyMap *	ags_frequency_map_new ()
#define	AGS_FREQUENCY_MAP()
#define	AGS_FREQUENCY_MAP_CLASS()

#define	AGS_FREQUENCY_MAP_GET_CLASS()
#define	AGS_IS_FREQUENCY_MAP()
#define	AGS_IS_FREQUENCY_MAP_CLASS()
GType	ags_frequency_map_get_type ()

Properties

guint	attack	Read / Write
guint	buffer-size	Read / Write
guint	frame-count	Read / Write
double	freq	Read / Write
guint	oscillator-mode	Read / Write
guint	samplerate	Read / Write
guint	window-count	Read / Write
guint	z-index	Read / Write

Signals

void	compute-max-likelihood	Run Last
void	factorize	Run Last
void	process	Run Last

Types and Values

#define	AGS_FREQUENCY_MAP_DEFAULT_FREQ
enum	AgsFrequencyMapFlags
#define	AGS_TYPE_FREQUENCY_MAP
struct	AgsFrequencyMap
struct	AgsFrequencyMapClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsFrequencyMap
```

Includes

```
#include <ags/audio/ags_frequency_map.h>
```

Description

AgsFrequencyMap stores a buffer of complex numbers. Representing an discrete sampled oscillator wave at a giving frequency.

Functions

AGS_FREQUENCY_MAP_GET_OBJ_MUTEX()

```
#define AGS_FREQUENCY_MAP_GET_OBJ_MUTEX(obj) (&(((AgsFrequencyMap *) obj)->obj_mutex))
```

ags_frequency_map_sort_func ()

```
gint
ags_frequency_map_sort_func (gconstpointer a,
                             gconstpointer b);
```

Sort frequency maps.

Parameters

a	an AgsFrequencyMap	
b	an AgsFrequencyMap	

Returns

0 if equal, -1 if smaller and 1 if bigger offset

Since: 3.0.0

ags_frequency_map_test_flags ()

```
gboolean
ags_frequency_map_test_flags (AgsFrequencyMap *frequency_map,
                              guint flags);
```

Test *flags* to be set on *recall* .

Parameters

frequency_map	the AgsFrequencyMap	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_frequency_map_set_flags ()

```
void
ags_frequency_map_set_flags (AgsFrequencyMap *frequency_map,
                             guint flags);
```

Set flags.

Parameters

frequency_map	the AgsFrequencyMap	
flags	the flags	

Since: 3.0.0

ags_frequency_map_unset_flags ()

```
void
ags_frequency_map_unset_flags (AgsFrequencyMap *frequency_map,
                               guint flags);
```

Unset flags.

Parameters

frequency_map	the AgsFrequencyMap	
flags	the flags	

Since: 3.0.0

ags_frequency_map_process ()

```
void
ags_frequency_map_process (AgsFrequencyMap *frequency_map);
```

Process *frequency_map* .

Parameters

frequency_map	the AgsFrequencyMap	

Since: 3.0.0

ags_frequency_map_factorize ()

```
void
ags_frequency_map_factorize (AgsFrequencyMap *frequency_map,
                             AgsFrequencyMap *factorized_frequency_map);
```

Factorize *frequency_map* .

Parameters

frequency_map	the AgsFrequencyMap	
factorized_frequency_map	the factorized AgsFrequencyMap	

Since: 3.0.0

ags_frequency_map_compute_max_likelihood ()

```
void
ags_frequency_map_compute_max_likelihood
                               (AgsFrequencyMap *frequency_map,
                                AgsComplex *source,
                                AgsComplex **retval);
```

Compute maximum likelihood *source* and store in *retval* .

Parameters

frequency_map	the AgsFrequencyMap	
source	the source as AgsComplex buffer	
retval	the return location of result as AgsComplex buffer	

Since: 3.0.0

ags_frequency_map_new ()

```
AgsFrequencyMap~*
ags_frequency_map_new ();
```

Returns

a new AgsFrequencyMap

Since: 3.0.0

AGS_FREQUENCY_MAP()

```
#define AGS_FREQUENCY_MAP(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FREQUENCY_MAP, AgsFrequencyMap))
```

AGS_FREQUENCY_MAP_CLASS()

```
#define AGS_FREQUENCY_MAP_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FREQUENCY_MAP, AgsFrequencyMapClass))
```

AGS_FREQUENCY_MAP_GET_CLASS()

```
#define AGS_FREQUENCY_MAP_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FREQUENCY_MAP, AgsFrequencyMapClass))
```

AGS_IS_FREQUENCY_MAP()

```
#define AGS_IS_FREQUENCY_MAP(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FREQUENCY_MAP))
```

AGS_IS_FREQUENCY_MAP_CLASS()

```
#define AGS_IS_FREQUENCY_MAP_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FREQUENCY_MAP))
```

ags_frequency_map_get_type ()

```
GType
ags_frequency_map_get_type ();
```

Types and Values

AGS_FREQUENCY_MAP_DEFAULT_FREQ

```
#define AGS_FREQUENCY_MAP_DEFAULT_FREQ (440.0)
```

enum AgsFrequencyMapFlags

Members

AGS_FREQUENCY_MAP_PROCESSED	the frequency map was processed

AGS_TYPE_FREQUENCY_MAP

```
#define AGS_TYPE_FREQUENCY_MAP                (ags_frequency_map_get_type())
```

struct AgsFrequencyMap

```
struct AgsFrequencyMap;
```

struct AgsFrequencyMapClass

```
struct AgsFrequencyMapClass {
  GObjectClass gobject;

  void (*process)(AgsFrequencyMap *frequency_map);

  void (*factorize)(AgsFrequencyMap *frequency_map,
        AgsFrequencyMap *factorized_frequency_map);

  void (*compute_max_likelihood)(AgsFrequencyMap *frequency_map,
         AgsComplex *source,
         AgsComplex **retval);
};
```

Property Details

The "attack" property

```
"attack"                    guint
```

The attack to be used.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size to be used.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "frame-count" property

```
"frame-count"               guint
```

The initial size of audio data.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "freq" property

```
"freq"                      double
```

The freq to be used.

Owner: AgsFrequencyMap

Flags: Read / Write

Allowed values: >= -1

Default value: 440

Since: 3.0.0

The "oscillator-mode" property

```
  "oscillator-mode"           guint
```

The oscillator mode to be used.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "samplerate" property

```
  "samplerate"                guint
```

The samplerate to be used.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "window-count" property

```
  "window-count"              guint
```

The the window count of total z-indexes.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "z-index" property

```
  "z-index"                   guint
```

The z-index as position of this frequency map.

Owner: AgsFrequencyMap

Flags: Read / Write

Default value: 0

Since: 3.0.0

Signal Details

The "compute-max-likelihood" signal

```
void
user_function (AgsFrequencyMap *frequency_map,
               gpointer         source,
               gpointer         retval,
               gpointer         user_data)
```

The ::compute-max-likelihood signal notifies about max likelihooded commputed.

Parameters

frequency_map	the AgsFrequencyMap	
source	the source buffer	
retval	return location pointer to buffer	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "factorize" signal

```
void
user_function (AgsFrequencyMap *frequency_map,
               GObject         *factorized_frequency_map,
               gpointer         user_data)
```

The ::factorize signal notifies about factorizing map.

Parameters

frequency_map	the AgsFrequencyMap	
factorized_frequency_map	the factorized AgsFrequencyMap	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "process" signal

```
void
user_function (AgsFrequencyMap *frequency_map,
               gpointer         user_data)
```

The ::process signal notifies about processed buffer.

Parameters

frequency_map	the AgsFrequencyMap	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 8

Audio sinks and sources

Here are objects listed to interact your sound devices using ALSA, WASAPI, JACK, Pulseaudio or CoreAudio.

8.1 AgsFifoout

AgsFifoout — Output to pipe

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FIFOOUT_GET_OBJ_MUTEX()
GQuark	ags_fifoout_error_quark ()
gboolean	ags_fifoout_test_flags ()
void	ags_fifoout_set_flags ()
void	ags_fifoout_unset_flags ()
void	ags_fifoout_switch_buffer_flag ()
void	ags_fifoout_adjust_delay_and_attack ()
void	ags_fifoout_realloc_buffer ()
AgsFifoout *	ags_fifoout_new ()
#define	AGS_FIFOOUT()
#define	AGS_FIFOOUT_CLASS()
#define	AGS_FIFOOUT_GET_CLASS()
#define	AGS_IS_FIFOOUT()
#define	AGS_IS_FIFOOUT_CLASS()
GType	ags_fifoout_flags_get_type ()
GType	ags_fifoout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write

double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_FIFOOUT_DEFAULT_DEVICE
#define	AGS_FIFOOUT_DEFAULT_APP_BUFFER_SIZE
#define	AGS_FIFOOUT_DEFAULT_RING_BUFFER_SIZE
enum	AgsFifooutFlags
enum	AgsFifooutAppBufferMode
#define	AGS_FIFOOUT_ERROR
enum	AgsFifooutError
#define	AGS_TYPE_FIFOOUT
struct	AgsFifoout
struct	AgsFifooutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsFifooutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsFifoout
```

Implemented Interfaces

AgsFifoout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/ags_fifoout.h>
```

Description

AgsFifoout represents a pipe and supports output.

Functions

AGS_FIFOOUT_GET_OBJ_MUTEX()

```
#define AGS_FIFOOUT_GET_OBJ_MUTEX(obj) (&(((AgsFifoout *) obj)->obj_mutex))
```

ags_fifoout_error_quark ()

```
GQuark
ags_fifoout_error_quark ();
```

ags_fifoout_test_flags ()

```
gboolean
ags_fifoout_test_flags (AgsFifoout *fifoout,
                        guint flags);
```

Test *flags* to be set on *fifoout* .

Parameters

fifoout	the AgsFifoout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_fifoout_set_flags ()

```
void
ags_fifoout_set_flags (AgsFifoout *fifoout,
                       guint flags);
```

Enable a feature of *fifoout* .

Parameters

fifoout	the AgsFifoout	
flags	see AgsFifooutFlags	

Since: 3.0.0

ags_fifoout_unset_flags ()

```
void
ags_fifoout_unset_flags (AgsFifoout *fifoout,
                         guint flags);
```

Disable a feature of *fifoout* .

Parameters

fifoout	the AgsFifoout	
flags	see AgsFifooutFlags	

Since: 3.0.0

ags_fifoout_switch_buffer_flag ()

```
void
ags_fifoout_switch_buffer_flag (AgsFifoout *fifoout);
```

The buffer flag indicates the currently played buffer.

Parameters

fifoout	the AgsFifoout

Since: 3.0.0

ags_fifoout_adjust_delay_and_attack ()

```
void
ags_fifoout_adjust_delay_and_attack (AgsFifoout *fifoout);
```

Calculate delay and attack and reset it.

Parameters

fifoout	the AgsFifoout

Since: 3.0.0

ags_fifoout_realloc_buffer ()

```
void
ags_fifoout_realloc_buffer (AgsFifoout *fifoout);
```

Reallocate the internal audio buffer.

Parameters

fifoout	the AgsFifoout

Since: 3.0.0

ags_fifoout_new ()

```
AgsFifoout~*
ags_fifoout_new ();
```

Creates a new instance of AgsFifoout.

Returns

the new AgsFifoout

Since: 3.0.0

AGS_FIFOOUT()

```
#define AGS_FIFOOUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_FIFOOUT ↩
    , AgsFifoout))
```

AGS_FIFOOUT_CLASS()

```
#define AGS_FIFOOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class, AGS_TYPE_FIFOOUT,  ↩
    AgsFifoout))
```

AGS_FIFOOUT_GET_CLASS()

```
#define AGS_FIFOOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj, AGS_TYPE_FIFOOUT,  ↩
    AgsFifooutClass))
```

AGS_IS_FIFOOUT()

```
#define AGS_IS_FIFOOUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FIFOOUT))
```

AGS_IS_FIFOOUT_CLASS()

```
#define AGS_IS_FIFOOUT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_FIFOOUT ↩
    ))
```

ags_fifoout_flags_get_type ()

```
GType
ags_fifoout_flags_get_type ();
```

ags_fifoout_get_type ()

```
GType
ags_fifoout_get_type ();
```

Types and Values

AGS_FIFOOUT_DEFAULT_DEVICE

```
#define AGS_FIFOOUT_DEFAULT_DEVICE "/dev/null"
```

AGS_FIFOOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_FIFOOUT_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_FIFOOUT_DEFAULT_RING_BUFFER_SIZE

```
#define AGS_FIFOOUT_DEFAULT_RING_BUFFER_SIZE (8)
```

enum AgsFifooutFlags

Enum values to control the behavior or indicate internal state of AgsFifoout by enable/disable as flags.

Members

AGS_FIFOOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_FIFOOUT_START_PLAY	playback start-ing
AGS_FIFOOUT_PLAY	use first at-tack, in-stead of sec-ond one
AGS_FIFOOUT_SHUTDOWN	stop play-back
AGS_FIFOOUT_NONBLOCKING	do non-blocking calls
AGS_FIFOOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsFifooutAppBufferMode

Enum values to indicate internal state of AgsFifoout application buffer by setting mode.

Members

AGS_FIFOOUT_APP_BUFFER_0	ring-buffer 0
AGS_FIFOOUT_APP_BUFFER_1	ring-buffer 1
AGS_FIFOOUT_APP_BUFFER_2	ring-buffer 2
AGS_FIFOOUT_APP_BUFFER_3	ring-buffer 3

AGS_FIFOOUT_ERROR

```
#define AGS_FIFOOUT_ERROR (ags_fifoout_error_quark())
```

enum AgsFifooutError

Members

AGS_FIFOOUT_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_FIFOOUT

```
#define AGS_TYPE_FIFOOUT                (ags_fifoout_get_type())
```

struct AgsFifoout

```
struct AgsFifoout;
```

struct AgsFifooutClass

```
struct AgsFifooutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsFifoout

Flags: Read

Since: 3.0.0

The "bpm" property

```
"bpm"                    double
```

Beats per minute

Owner: AgsFifoout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
"buffer"                 gpointer
```

The buffer

Owner: AgsFifoout

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"            guint
```

The buffer size

Owner: AgsFifoout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"           double
```

tact

Owner: AgsFifoout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The fifo soundcard indentifier

Owner: AgsFifoout

Flags: Read / Write

Default value: "/dev/null"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsFifoout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsFifoout

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsFifoout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsFifoout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.0.0

8.2 AgsAlsaDevin

AgsAlsaDevin — Input from soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_ALSA_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_alsa_devin_error_quark ()
gboolean	ags_alsa_devin_test_flags ()
void	ags_alsa_devin_set_flags ()
void	ags_alsa_devin_unset_flags ()
void	ags_alsa_devin_switch_buffer_flag ()
void	ags_alsa_devin_adjust_delay_and_attack ()
void	ags_alsa_devin_realloc_buffer ()
AgsAlsaDevin *	ags_alsa_devin_new ()
#define	AGS_ALSA_DEVIN()
#define	AGS_ALSA_DEVIN_CLASS()
#define	AGS_ALSA_DEVIN_GET_CLASS()
#define	AGS_IS_ALSA_DEVIN()
#define	AGS_IS_ALSA_DEVIN_CLASS()
GType	ags_alsa_devin_flags_get_type ()
GType	ags_alsa_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write

guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_ALSA_DEVIN_DEFAULT_ALSA_DEVICE
#define	AGS_ALSA_DEVIN_DEFAULT_APP_BUFFER_SIZE
#define	AGS_ALSA_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE
enum	AgsAlsaDevinFlags
enum	AgsAlsaDevinAppBufferMode
enum	AgsAlsaDevinBackendBufferMode
#define	AGS_ALSA_DEVIN_ERROR
enum	AgsAlsaDevinError
#define	AGS_TYPE_ALSA_DEVIN
struct	AgsAlsaDevin
struct	AgsAlsaDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsAlsaDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsAlsaDevin
```

Implemented Interfaces

AgsAlsaDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/ags_alsa_devin.h>
```

Description

AgsAlsaDevin represents a soundcard and supports input.

Functions

AGS_ALSA_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_ALSA_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsAlsaDevin *) obj)->obj_mutex))
```

ags_alsa_devin_error_quark ()

```
GQuark
ags_alsa_devin_error_quark ();
```

ags_alsa_devin_test_flags ()

```
gboolean
ags_alsa_devin_test_flags (AgsAlsaDevin *alsa_devin,
                           guint flags);
```

Test *flags* to be set on *alsa_devin* .

Parameters

alsa_devin	the AgsAlsaDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.13.2

ags_alsa_devin_set_flags ()

```
void
ags_alsa_devin_set_flags (AgsAlsaDevin *alsa_devin,
                          guint flags);
```

Enable a feature of *alsa_devin* .

Parameters

alsa_devin	the AgsAlsaDevin	
flags	see AgsAlsaDevinFlags	

Since: 3.13.2

ags_alsa_devin_unset_flags ()

```
void
ags_alsa_devin_unset_flags (AgsAlsaDevin *alsa_devin,
                            guint flags);
```

Disable a feature of *alsa_devin* .

Parameters

alsa_devin	the AgsAlsaDevin	
flags	see AgsAlsaDevinFlags	

Since: 3.13.2

ags_alsa_devin_switch_buffer_flag ()

```
void
ags_alsa_devin_switch_buffer_flag (AgsAlsaDevin *alsa_devin);
```

The buffer flag indicates the currently recorded buffer.

Parameters

alsa_devin	the AgsAlsaDevin

Since: 3.13.2

ags_alsa_devin_adjust_delay_and_attack ()

```
void
ags_alsa_devin_adjust_delay_and_attack
                                (AgsAlsaDevin *alsa_devin);
```

Calculate delay and attack and reset it.

Parameters

alsa_devin	the AgsAlsaDevin

Since: 3.13.2

ags_alsa_devin_realloc_buffer ()

```
void
ags_alsa_devin_realloc_buffer (AgsAlsaDevin *alsa_devin);
```

Reallocate the internal audio buffer.

Parameters

alsa_devin	the AgsAlsaDevin

Since: 3.13.2

ags_alsa_devin_new ()

```
AgsAlsaDevin~*
ags_alsa_devin_new ();
```

Creates a new instance of AgsAlsaDevin.

Returns

the new AgsAlsaDevin

Since: 3.13.2

AGS_ALSA_DEVIN()

```
#define AGS_ALSA_DEVIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ALSA_DEVIN, AgsAlsaDevin))
```

AGS_ALSA_DEVIN_CLASS()

```
#define AGS_ALSA_DEVIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_ALSA_DEVIN, AgsAlsaDevin))
```

AGS_ALSA_DEVIN_GET_CLASS()

```
#define AGS_ALSA_DEVIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_ALSA_DEVIN, AgsAlsaDevinClass))
```

AGS_IS_ALSA_DEVIN()

```
#define AGS_IS_ALSA_DEVIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_ALSA_DEVIN))
```

AGS_IS_ALSA_DEVIN_CLASS()

```
#define AGS_IS_ALSA_DEVIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_ALSA_DEVIN))
```

ags_alsa_devin_flags_get_type ()

```
GType
ags_alsa_devin_flags_get_type ();
```

ags_alsa_devin_get_type ()

```
GType
ags_alsa_devin_get_type ();
```

Types and Values

AGS_ALSA_DEVIN_DEFAULT_ALSA_DEVICE

```
#define AGS_ALSA_DEVIN_DEFAULT_ALSA_DEVICE "hw:0,0"
```

AGS_ALSA_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_ALSA_DEVIN_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_ALSA_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE

```
#define AGS_ALSA_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE (8)
```

enum AgsAlsaDevinFlags

Enum values to control the behavior or indicate internal state of AgsAlsaDevin by enable/disable as flags.

Members

AGS_ALSA_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_ALSA_DEVIN_START_RECORD	capture start-ing
AGS_ALSA_DEVIN_RECORD	do cap-ture
AGS_ALSA_DEVIN_SHUTDOWN	stop cap-ture
AGS_ALSA_DEVIN_NONBLOCKING	do non-blocking calls
AGS_ALSA_DEVIN_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsAlsaDevinAppBufferMode

Enum values to indicate internal state of AgsAlsaDevin application buffer by setting mode.

Members

AGS_ALSA_DEVIN_APP_BUFFER_0	ring-buffer 0

AGS_ALSA_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_ALSA_DEVIN_APP_BUFFER_2	ring-buffer 2
AGS_ALSA_DEVIN_APP_BUFFER_3	ring-buffer 3

enum AgsAlsaDevinBackendBufferMode

Enum values to indicate internal state of AgsAlsaDevin backend buffer by setting mode.

Members

AGS_ALSA_DEVIN_BACKEND_BUFFER_0	ring-buffer 0
AGS_ALSA_DEVIN_BACKEND_BUFFER_1	ring-buffer 1
AGS_ALSA_DEVIN_BACKEND_BUFFER_2	ring-buffer 2
AGS_ALSA_DEVIN_BACKEND_BUFFER_3	ring-buffer 3
AGS_ALSA_DEVIN_BACKEND_BUFFER_4	ring-buffer 4
AGS_ALSA_DEVIN_BACKEND_BUFFER_5	ring-buffer 5
AGS_ALSA_DEVIN_BACKEND_BUFFER_6	ring-buffer 6
AGS_ALSA_DEVIN_BACKEND_BUFFER_7	ring-buffer 7

AGS_ALSA_DEVIN_ERROR

```
#define AGS_ALSA_DEVIN_ERROR (ags_alsa_devin_error_quark())
```

enum AgsAlsaDevinError

Members

AGS_ALSA_DEVIN_ERROR_LOCKED_SOUNDCARD		
AGS_ALSA_DEVIN_ERROR_BROKEN_CONFIGURATION		
AGS_ALSA_DEVIN_ERROR_ACCESS_TYPE_NOT_AVAILABLE		

AGS_ALSA_DEVIN_ERROR_SAMPLE_FORMAT_NOT_AVAILABLE		
AGS_ALSA_DEVIN_ERROR_CHANNELS_NOT_AVAILABLE		
AGS_ALSA_DEVIN_ERROR_SAMPLERATE_NOT_AVAILABLE		
AGS_ALSA_DEVIN_ERROR_BUFFER_SIZE_NOT_AVAILABLE		
AGS_ALSA_DEVIN_ERROR_HW_PARAMETERS_NOT_AVAILABLE		

AGS_TYPE_ALSA_DEVIN

```
#define AGS_TYPE_ALSA_DEVIN                (ags_alsa_devin_get_type())
```

struct AgsAlsaDevin

```
struct AgsAlsaDevin;
```

struct AgsAlsaDevinClass

```
struct AgsAlsaDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsAlsaDevin

Flags: Read

Since: 3.13.2

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsAlsaDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.13.2

The "buffer" property

```
"buffer"                    gpointer
```

The buffer

Owner: AgsAlsaDevin

Flags: Read

Since: 3.13.2

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size

Owner: AgsAlsaDevin

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.13.2

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsAlsaDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.13.2

The "device" property

```
"device"                    char~*
```

The alsa soundcard indentifier

Owner: AgsAlsaDevin

Flags: Read / Write

Default value: "hw:0,0"

Since: 3.13.2

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsAlsaDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsAlsaDevin

Flags: Read / Write

Default value: 16

Since: 3.13.2

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsAlsaDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsAlsaDevin

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.13.2

8.3 AgsAlsaDevout

AgsAlsaDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_ALSA_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_alsa_devout_error_quark ()
gboolean	ags_alsa_devout_test_flags ()
void	ags_alsa_devout_set_flags ()
void	ags_alsa_devout_unset_flags ()
void	ags_alsa_devout_switch_buffer ()
void	ags_alsa_devout_adjust_delay_and_attack ()
void	ags_alsa_devout_realloc_buffer ()
AgsAlsaDevout *	ags_alsa_devout_new ()
#define	AGS_ALSA_DEVOUT()
#define	AGS_ALSA_DEVOUT_CLASS()
#define	AGS_ALSA_DEVOUT_GET_CLASS()
#define	AGS_IS_ALSA_DEVOUT()
#define	AGS_IS_ALSA_DEVOUT_CLASS()
GType	ags_alsa_devout_flags_get_type ()
GType	ags_alsa_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_ALSA_DEVOUT_DEFAULT_ALSA_DEVICE
#define	AGS_ALSA_DEVOUT_DEFAULT_APP_BUFFER_SIZE
#define	AGS_ALSA_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE
enum	AgsAlsaDevoutFlags
enum	AgsAlsaDevoutAppBufferMode
enum	AgsAlsaDevoutBackendBufferMode
#define	AGS_ALSA_DEVOUT_ERROR
enum	AgsAlsaDevoutError
#define	AGS_TYPE_ALSA_DEVOUT

struct	AgsAlsaDevout
struct	AgsAlsaDevoutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsAlsaDevoutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsAlsaDevout
```

Implemented Interfaces

AgsAlsaDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/ags_alsa_devout.h>
```

Description

AgsAlsaDevout represents a soundcard and supports output.

Functions

AGS_ALSA_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_ALSA_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsAlsaDevout *) obj)->obj_mutex))
```

ags_alsa_devout_error_quark ()

```
GQuark
ags_alsa_devout_error_quark ();
```

ags_alsa_devout_test_flags ()

```
gboolean
ags_alsa_devout_test_flags (AgsAlsaDevout *alsa_devout,
                            guint flags);
```

Test *flags* to be set on *alsa_devout* .

Parameters

alsa_devout	the AgsAlsaDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.13.2

ags_alsa_devout_set_flags ()

```
void
ags_alsa_devout_set_flags (AgsAlsaDevout *alsa_devout,
                           guint flags);
```

Enable a feature of *alsa_devout* .

Parameters

alsa_devout	the AgsAlsaDevout	
flags	see AgsAlsaDevoutFlags	

Since: 3.13.2

ags_alsa_devout_unset_flags ()

```
void
ags_alsa_devout_unset_flags (AgsAlsaDevout *alsa_devout,
                             guint flags);
```

Disable a feature of *alsa_devout* .

Parameters

alsa_devout	the AgsAlsaDevout	
flags	see AgsAlsaDevoutFlags	

Since: 3.13.2

ags_alsa_devout_switch_buffer ()

```
void
ags_alsa_devout_switch_buffer (AgsAlsaDevout *alsa_devout);
```

Switch currently played buffer.

Parameters

alsa_devout	the AgsAlsaDevout	

Since: 3.13.2

ags_alsa_devout_adjust_delay_and_attack ()

```
void
ags_alsa_devout_adjust_delay_and_attack
                                (AgsAlsaDevout *alsa_devout);
```

Calculate delay and attack and reset it.

Parameters

alsa_devout	the AgsAlsaDevout

Since: 3.13.2

ags_alsa_devout_realloc_buffer ()

```
void
ags_alsa_devout_realloc_buffer (AgsAlsaDevout *alsa_devout);
```

Reallocate the internal audio buffer.

Parameters

alsa_devout	the AgsAlsaDevout

Since: 3.13.2

ags_alsa_devout_new ()

```
AgsAlsaDevout~*
ags_alsa_devout_new ();
```

Creates a new instance of AgsAlsaDevout.

Returns

the new AgsAlsaDevout

Since: 3.13.2

AGS_ALSA_DEVOUT()

```
#define AGS_ALSA_DEVOUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ALSA_DEVOUT, AgsAlsaDevout))
```

AGS_ALSA_DEVOUT_CLASS()

```
#define AGS_ALSA_DEVOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_ALSA_DEVOUT, AgsAlsaDevout))
```

AGS_ALSA_DEVOUT_GET_CLASS()

```
#define AGS_ALSA_DEVOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_ALSA_DEVOUT, AgsAlsaDevoutClass))
```

AGS_IS_ALSA_DEVOUT()

```
#define AGS_IS_ALSA_DEVOUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_ALSA_DEVOUT))
```

AGS_IS_ALSA_DEVOUT_CLASS()

```
#define AGS_IS_ALSA_DEVOUT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_ALSA_DEVOUT))
```

ags_alsa_devout_flags_get_type ()

```
GType
ags_alsa_devout_flags_get_type ();
```

ags_alsa_devout_get_type ()

```
GType
ags_alsa_devout_get_type ();
```

Types and Values

AGS_ALSA_DEVOUT_DEFAULT_ALSA_DEVICE

```
#define AGS_ALSA_DEVOUT_DEFAULT_ALSA_DEVICE "hw:0,0"
```

AGS_ALSA_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_ALSA_DEVOUT_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_ALSA_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE

```
#define AGS_ALSA_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE (8)
```

enum AgsAlsaDevoutFlags

Enum values to control the behavior or indicate internal state of AgsAlsaDevout by enable/disable as flags.

Members

AGS_ALSA_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_ALSA_DEVOUT_START_PLAY	playback start-ing
AGS_ALSA_DEVOUT_PLAY	devout is run-ning
AGS_ALSA_DEVOUT_SHUTDOWN	stop play-back
AGS_ALSA_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_ALSA_DEVOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsAlsaDevoutAppBufferMode

Enum values to indicate internal state of AgsAlsaDevout application buffer by setting mode.

Members

AGS_ALSA_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_ALSA_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_ALSA_DEVOUT_APP_BUFFER_2	ring-buffer 2
AGS_ALSA_DEVOUT_APP_BUFFER_3	ring-buffer 3

enum AgsAlsaDevoutBackendBufferMode

Enum values to indicate internal state of AgsAlsaDevout backend buffer by setting mode.

Members

AGS_ALSA_DEVOUT_BACKEND_BUFFER_0	ring-buffer 0
AGS_ALSA_DEVOUT_BACKEND_BUFFER_1	ring-buffer 1
AGS_ALSA_DEVOUT_BACKEND_BUFFER_2	ring-buffer 2
AGS_ALSA_DEVOUT_BACKEND_BUFFER_3	ring-buffer 3
AGS_ALSA_DEVOUT_BACKEND_BUFFER_4	ring-buffer 4
AGS_ALSA_DEVOUT_BACKEND_BUFFER_5	ring-buffer 5
AGS_ALSA_DEVOUT_BACKEND_BUFFER_6	ring-buffer 6
AGS_ALSA_DEVOUT_BACKEND_BUFFER_7	ring-buffer 7

AGS_ALSA_DEVOUT_ERROR

```
#define AGS_ALSA_DEVOUT_ERROR (ags_alsa_devout_error_quark())
```

enum AgsAlsaDevoutError

Members

AGS_ALSA_DEVOUT_ERROR_LOCKED_SOUNDCARD	
AGS_ALSA_DEVOUT_ERROR_BROKEN_CONFIGURATION	
AGS_ALSA_DEVOUT_ERROR_ACCESS_TYPE_NOT_AVAILABLE	
AGS_ALSA_DEVOUT_ERROR_SAMPLE_FORMAT_NOT_AVAILABLE	
AGS_ALSA_DEVOUT_ERROR_CHANNELS_NOT_AVAILABLE	
AGS_ALSA_DEVOUT_ERROR_SAMPLERATE_NOT_AVAILABLE	
AGS_ALSA_DEVOUT_ERROR_BUFFER_SIZE_NOT_AVAILABLE	
AGS_ALSA_DEVOUT_ERROR_HW_PARAMETERS_NOT_AVAILABLE	

AGS_TYPE_ALSA_DEVOUT

```
#define AGS_TYPE_ALSA_DEVOUT                (ags_alsa_devout_get_type())
```

struct AgsAlsaDevout

```
struct AgsAlsaDevout;
```

struct AgsAlsaDevoutClass

```
struct AgsAlsaDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsAlsaDevout

Flags: Read

Since: 3.13.2

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsAlsaDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.13.2

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsAlsaDevout

Flags: Read

Since: 3.13.2

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size

Owner: AgsAlsaDevout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.13.2

The "delay-factor" property

```
"delay-factor"               double
```

tact

Owner: AgsAlsaDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.13.2

The "device" property

```
"device"                     char~*
```

The alsa soundcard indentifier

Owner: AgsAlsaDevout

Flags: Read / Write

Default value: "hw:0,0"

Since: 3.13.2

The "dsp-channels" property

```
"dsp-channels"               guint
```

The dsp channel count

Owner: AgsAlsaDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsAlsaDevout

Flags: Read / Write

Default value: 16

Since: 3.13.2

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsAlsaDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsAlsaDevout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.13.2

8.4 AgsAlsaMidiin

AgsAlsaMidiin — Input from sequencer

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_ALSA_MIDIIN_GET_OBJ_MUTEX()
GQuark	ags_alsa_midiin_error_quark ()
gboolean	ags_alsa_midiin_test_flags ()
void	ags_alsa_midiin_set_flags ()
void	ags_alsa_midiin_unset_flags ()
void	ags_alsa_midiin_switch_buffer_flag ()
AgsAlsaMidiin *	ags_alsa_midiin_new ()
#define	AGS_ALSA_MIDIIN()
#define	AGS_ALSA_MIDIIN_CLASS()
#define	AGS_ALSA_MIDIIN_GET_CLASS()
#define	AGS_IS_ALSA_MIDIIN()
#define	AGS_IS_ALSA_MIDIIN_CLASS()
GType	ags_alsa_midiin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
double	delay-factor	Read / Write
char *	device	Read / Write

Types and Values

#define	AGS_ALSA_MIDIIN_DEFAULT_ALSA_DEVICE
#define	AGS_ALSA_MIDIIN_DEFAULT_APP_BUFFER_SIZE
#define	AGS_ALSA_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE
#define	AGS_ALSA_MIDIIN_DEFAULT_BUFFER_SIZE
enum	AgsAlsaMidiinFlags
enum	AgsAlsaMidiinAppBufferMode
enum	AgsAlsaMidiinBackendBufferMode
enum	AgsAlsaMidiinSyncFlags
#define	AGS_ALSA_MIDIIN_ERROR
enum	AgsAlsaMidiinError
#define	AGS_TYPE_ALSA_MIDIIN
struct	AgsAlsaMidiin
struct	AgsAlsaMidiinClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsAlsaMidiin
```

Implemented Interfaces

AgsAlsaMidiin implements AgsConnectable and AgsSequencer.

Includes

```
#include <ags/audio/ags_alsa_midiin.h>
```

Description

AgsAlsaMidiin represents a sequencer and supports midi input.

Functions

AGS_ALSA_MIDIIN_GET_OBJ_MUTEX()

```
#define AGS_ALSA_MIDIIN_GET_OBJ_MUTEX(obj) (&(((AgsAlsaMidiin *) obj)->obj_mutex))
```

ags_alsa_midiin_error_quark ()

```
GQuark
ags_alsa_midiin_error_quark ();
```

ags_alsa_midiin_test_flags ()

```
gboolean
ags_alsa_midiin_test_flags (AgsAlsaMidiin *alsa_midiin,
                            guint flags);
```

Test *flags* to be set on *alsa_midiin* .

Parameters

alsa_midiin	the AgsAlsaMidiin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.13.2

ags_alsa_midiin_set_flags ()

```
void
ags_alsa_midiin_set_flags (AgsAlsaMidiin *alsa_midiin,
                           guint flags);
```

Enable a feature of *alsa_midiin* .

Parameters

alsa_midiin	the AgsAlsaMidiin	
flags	see AgsAlsaMidiinFlags	

Since: 3.13.2

ags_alsa_midiin_unset_flags ()

```
void
ags_alsa_midiin_unset_flags (AgsAlsaMidiin *alsa_midiin,
                             guint flags);
```

Disable a feature of *alsa_midiin*.

Parameters

alsa_midiin	the AgsAlsaMidiin	
flags	see AgsAlsaMidiinFlags	

Since: 3.13.2

ags_alsa_midiin_switch_buffer_flag ()

```
void
ags_alsa_midiin_switch_buffer_flag (AgsAlsaMidiin *alsa_midiin);
```

The buffer flag indicates the currently played buffer.

Parameters

alsa_midiin	the AgsAlsaMidiin	

Since: 3.13.2

ags_alsa_midiin_new ()

```
AgsAlsaMidiin~*
ags_alsa_midiin_new ();
```

Creates a new instance of AgsAlsaMidiin.

Returns

the new AgsAlsaMidiin

Since: 3.13.2

AGS_ALSA_MIDIIN()

```
#define AGS_ALSA_MIDIIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ALSA_MIDIIN, AgsAlsaMidiin))
```

AGS_ALSA_MIDIIN_CLASS()

```
#define AGS_ALSA_MIDIIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_ALSA_MIDIIN, AgsAlsaMidiin))
```

AGS_ALSA_MIDIIN_GET_CLASS()

```
#define AGS_ALSA_MIDIIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_ALSA_MIDIIN, AgsAlsaMidiinClass))
```

AGS_IS_ALSA_MIDIIN()

```
#define AGS_IS_ALSA_MIDIIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_ALSA_MIDIIN))
```

AGS_IS_ALSA_MIDIIN_CLASS()

```
#define AGS_IS_ALSA_MIDIIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_ALSA_MIDIIN))
```

ags_alsa_midiin_get_type ()

```
GType
ags_alsa_midiin_get_type ();
```

Types and Values

AGS_ALSA_MIDIIN_DEFAULT_ALSA_DEVICE

```
#define AGS_ALSA_MIDIIN_DEFAULT_ALSA_DEVICE "hw:0,0"
```

AGS_ALSA_MIDIIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_ALSA_MIDIIN_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_ALSA_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE

```
#define AGS_ALSA_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE (8)
```

AGS_ALSA_MIDIIN_DEFAULT_BUFFER_SIZE

```
#define AGS_ALSA_MIDIIN_DEFAULT_BUFFER_SIZE (4096)
```

enum AgsAlsaMidiinFlags

Enum values to control the behavior or indicate internal state of AgsAlsaMidiin by enable/disable as flags.

Members

AGS_ALSA_MIDIIN_ADDED_TO_REGISTRY	the alsa_midiin was added to registry, see AgsConnectable::add_to_registry()
AGS_ALSA_MIDIIN_CONNECTED	indicates the alsa_midiin was connected by calling AgsConnectable::connect()
AGS_ALSA_MIDIIN_INITIALIZED	recording is initialized
AGS_ALSA_MIDIIN_RECORD	is recording
AGS_ALSA_MIDIIN_SHUTDOWN	stop recording
AGS_ALSA_MIDIIN_START_RECORD	just started recording
AGS_ALSA_MIDIIN_NONBLOCKING	do non-blocking calls
AGS_ALSA_MIDIIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsAlsaMidiinAppBufferMode

Enum values to indicate internal state of AgsAlsaMidiin application buffer by setting mode.

Members

AGS_ALSA_MIDIIN_APP_BUFFER_0	ring-buffer 0
AGS_ALSA_MIDIIN_APP_BUFFER_1	ring-buffer 1
AGS_ALSA_MIDIIN_APP_BUFFER_2	ring-buffer 2
AGS_ALSA_MIDIIN_APP_BUFFER_3	ring-buffer 3

enum AgsAlsaMidiinBackendBufferMode

Enum values to indicate internal state of AgsAlsaMidiin backend buffer by setting mode.

Members

AGS_ALSA_MIDIIN_BACKEND_BUFFER_0	ring-buffer 0
AGS_ALSA_MIDIIN_BACKEND_BUFFER_1	ring-buffer 1
AGS_ALSA_MIDIIN_BACKEND_BUFFER_2	ring-buffer 2
AGS_ALSA_MIDIIN_BACKEND_BUFFER_3	ring-buffer 3
AGS_ALSA_MIDIIN_BACKEND_BUFFER_4	ring-buffer 4
AGS_ALSA_MIDIIN_BACKEND_BUFFER_5	ring-buffer 5
AGS_ALSA_MIDIIN_BACKEND_BUFFER_6	ring-buffer 6
AGS_ALSA_MIDIIN_BACKEND_BUFFER_7	ring-buffer 7

enum AgsAlsaMidiinSyncFlags

Enum values to control the synchronization between sequencer and poll.

Members

AGS_ALSA_MIDIIN_PASS_THROUGH	do not sync
AGS_ALSA_MIDIIN_INITIAL_POLL	initial poll
AGS_ALSA_MIDIIN_POLL_WAIT	sync wait, sequencer conditional lock
AGS_ALSA_MIDIIN_POLL_DONE	sync done, sequencer conditional lock
AGS_ALSA_MIDIIN_POLL_FINISH_WAIT	sync wait, client conditional lock
AGS_ALSA_MIDIIN_POLL_FINISH_DONE	sync done, client conditional lock
AGS_ALSA_MIDIIN_POLL_SWITCH_BUFFER	switch buffer

AGS_ALSA_MIDIIN_ERROR

```
#define AGS_ALSA_MIDIIN_ERROR (ags_alsa_midiin_error_quark())
```

enum AgsAlsaMidiinError

Members

AGS_ALSA_MIDIIN_ERROR_LOCKED_SEQUENCER	

AGS_TYPE_ALSA_MIDIIN

```
#define AGS_TYPE_ALSA_MIDIIN                (ags_alsa_midiin_get_type())
```

struct AgsAlsaMidiin

```
struct AgsAlsaMidiin;
```

struct AgsAlsaMidiinClass

```
struct AgsAlsaMidiinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsAlsaMidiin

Flags: Read

Since: 3.13.2

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsAlsaMidiin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.13.2

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsAlsaMidiin

Flags: Read

Since: 3.13.2

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsAlsaMidiin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.13.2

The "device" property

```
"device"                    char~*
```

The alsa sequencer indentifier

Owner: AgsAlsaMidiin

Flags: Read / Write

Default value: "hw:0,0"

Since: 3.13.2

8.5 AgsOssDevin

AgsOssDevin — Input from soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSS_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_oss_devin_error_quark ()
gboolean	ags_oss_devin_test_flags ()
void	ags_oss_devin_set_flags ()
void	ags_oss_devin_unset_flags ()
void	ags_oss_devin_switch_buffer_flag ()
void	ags_oss_devin_adjust_delay_and_attack ()
void	ags_oss_devin_realloc_buffer ()
AgsOssDevin *	ags_oss_devin_new ()
#define	AGS_IS_OSS_DEVIN()
#define	AGS_IS_OSS_DEVIN_CLASS()
#define	AGS_OSS_DEVIN()
#define	AGS_OSS_DEVIN_CLASS()
#define	AGS_OSS_DEVIN_GET_CLASS()
GType	ags_oss_devin_flags_get_type ()
GType	ags_oss_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_OSS_DEVIN_DEFAULT_OSS_DEVICE
#define	AGS_OSS_DEVIN_DEFAULT_OSS_MIXER_DEVICE
#define	AGS_OSS_DEVIN_DEFAULT_APP_BUFFER_SIZE
#define	AGS_OSS_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE
enum	AgsOssDevinFlags
enum	AgsOssDevinAppBufferMode
enum	AgsOssDevinBackendBufferMode
#define	AGS_OSS_DEVIN_ERROR
enum	AgsOssDevinError
#define	AGS_TYPE_OSS_DEVIN
struct	AgsOssDevin
struct	AgsOssDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOssDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsOssDevin
```

Implemented Interfaces

AgsOssDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/ags_oss_devin.h>
```

Description

AgsOssDevin represents a soundcard and supports input.

Functions

AGS_OSS_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_OSS_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsOssDevin *) obj)->obj_mutex))
```

ags_oss_devin_error_quark ()

```
GQuark
ags_oss_devin_error_quark ();
```

ags_oss_devin_test_flags ()

```
gboolean
ags_oss_devin_test_flags (AgsOssDevin *oss_devin,
                          guint flags);
```

Test *flags* to be set on *oss_devin* .

Parameters

oss_devin	the AgsOssDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.13.2

ags_oss_devin_set_flags ()

```
void
ags_oss_devin_set_flags (AgsOssDevin *oss_devin,
                         guint flags);
```

Enable a feature of *oss_devin* .

Parameters

oss_devin	the AgsOssDevin	
flags	see AgsOssDevinFlags	

Since: 3.13.2

ags_oss_devin_unset_flags ()

```
void
ags_oss_devin_unset_flags (AgsOssDevin *oss_devin,
                           guint flags);
```

Disable a feature of *oss_devin* .

Parameters

oss_devin	the AgsOssDevin	
flags	see AgsOssDevinFlags	

Since: 3.13.2

ags_oss_devin_switch_buffer_flag ()

```
void
ags_oss_devin_switch_buffer_flag (AgsOssDevin *oss_devin);
```

The buffer flag indicates the currently recorded buffer.

Parameters

oss_devin	the AgsOssDevin	

Since: 3.13.2

ags_oss_devin_adjust_delay_and_attack ()

```
void
ags_oss_devin_adjust_delay_and_attack (AgsOssDevin *oss_devin);
```

Calculate delay and attack and reset it.

Parameters

oss_devin	the AgsOssDevin	

Since: 3.13.2

ags_oss_devin_realloc_buffer ()

```
void
ags_oss_devin_realloc_buffer (AgsOssDevin *oss_devin);
```

Reallocate the internal audio buffer.

Parameters

oss_devin	the AgsOssDevin	

Since: 3.13.2

ags_oss_devin_new ()

```
AgsOssDevin~*
ags_oss_devin_new ();
```

Creates a new instance of AgsOssDevin.

Returns

the new AgsOssDevin

Since: 3.13.2

AGS_IS_OSS_DEVIN()

```
#define AGS_IS_OSS_DEVIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSS_DEVIN))
```

AGS_IS_OSS_DEVIN_CLASS()

```
#define AGS_IS_OSS_DEVIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSS_DEVIN))
```

AGS_OSS_DEVIN()

```
#define AGS_OSS_DEVIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSS_DEVIN, AgsOssDevin))
```

AGS_OSS_DEVIN_CLASS()

```
#define AGS_OSS_DEVIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSS_DEVIN, AgsOssDevin))
```

AGS_OSS_DEVIN_GET_CLASS()

```
#define AGS_OSS_DEVIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSS_DEVIN, AgsOssDevinClass))
```

ags_oss_devin_flags_get_type ()

```
GType
ags_oss_devin_flags_get_type ();
```

ags_oss_devin_get_type ()

```
GType
ags_oss_devin_get_type ();
```

Types and Values

AGS_OSS_DEVIN_DEFAULT_OSS_DEVICE

```
#define AGS_OSS_DEVIN_DEFAULT_OSS_DEVICE "/dev/dsp"
```

AGS_OSS_DEVIN_DEFAULT_OSS_MIXER_DEVICE

```
#define AGS_OSS_DEVIN_DEFAULT_OSS_MIXER_DEVICE "/dev/mixer"
```

AGS_OSS_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_OSS_DEVIN_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_OSS_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE

```
#define AGS_OSS_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE (8)
```

enum AgsOssDevinFlags

Enum values to control the behavior or indicate internal state of AgsOssDevin by enable/disable as flags.

Members

AGS_OSS_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_OSS_DEVIN_START_RECORD	capture start-ing
AGS_OSS_DEVIN_RECORD	do cap-ture
AGS_OSS_DEVIN_SHUTDOWN	stop cap-ture
AGS_OSS_DEVIN_NONBLOCKING	do non-blocking calls

AGS_OSS_DEVIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsOssDevinAppBufferMode

Enum values to indicate internal state of AgsOssDevin application buffer by setting mode.

Members

AGS_OSS_DEVIN_APP_BUFFER_0	ring-buffer 0
AGS_OSS_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_OSS_DEVIN_APP_BUFFER_2	ring-buffer 2
AGS_OSS_DEVIN_APP_BUFFER_3	ring-buffer 3

enum AgsOssDevinBackendBufferMode

Enum values to indicate internal state of AgsOssDevin backend buffer by setting mode.

Members

AGS_OSS_DEVIN_BACKEND_BUFFER_0	ring-buffer 0
AGS_OSS_DEVIN_BACKEND_BUFFER_1	ring-buffer 1
AGS_OSS_DEVIN_BACKEND_BUFFER_2	ring-buffer 2
AGS_OSS_DEVIN_BACKEND_BUFFER_3	ring-buffer 3
AGS_OSS_DEVIN_BACKEND_BUFFER_4	ring-buffer 4

AGS_OSS_DEVIN_BACKEND_BUFFER_5	ring-buffer 5
AGS_OSS_DEVIN_BACKEND_BUFFER_6	ring-buffer 6
AGS_OSS_DEVIN_BACKEND_BUFFER_7	ring-buffer 7

AGS_OSS_DEVIN_ERROR

```
#define AGS_OSS_DEVIN_ERROR (ags_oss_devin_error_quark())
```

enum AgsOssDevinError

Members

AGS_OSS_DEVIN_ERROR_LOCKED_SOUNDCARD		
AGS_OSS_DEVIN_ERROR_BROKEN_CONFIGURATION		
AGS_OSS_DEVIN_ERROR_ACCESS_TYPE_NOT_AVAILABLE		
AGS_OSS_DEVIN_ERROR_SAMPLE_FORMAT_NOT_AVAILABLE		
AGS_OSS_DEVIN_ERROR_CHANNELS_NOT_AVAILABLE		
AGS_OSS_DEVIN_ERROR_SAMPLERATE_NOT_AVAILABLE		
AGS_OSS_DEVIN_ERROR_BUFFER_SIZE_NOT_AVAILABLE		
AGS_OSS_DEVIN_ERROR_HW_PARAMETERS_NOT_AVAILABLE		

AGS_TYPE_OSS_DEVIN

```
#define AGS_TYPE_OSS_DEVIN                (ags_oss_devin_get_type())
```

struct AgsOssDevin

```
struct AgsOssDevin;
```

struct AgsOssDevinClass

```
struct AgsOssDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsOssDevin

Flags: Read

Since: 3.13.2

The "bpm" property

```
"bpm"                    double
```

Beats per minute

Owner: AgsOssDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.13.2

The "buffer" property

```
"buffer"                 gpointer
```

The buffer

Owner: AgsOssDevin

Flags: Read

Since: 3.13.2

The "buffer-size" property

```
"buffer-size"            guint
```

The buffer size

Owner: AgsOssDevin

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.13.2

The "delay-factor" property

```
"delay-factor"           double
```

tact

Owner: AgsOssDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.13.2

The "device" property

```
"device"                    char~*
```

The soundcard indentifier

Owner: AgsOssDevin

Flags: Read / Write

Default value: "/dev/dsp"

Since: 3.13.2

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsOssDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsOssDevin

Flags: Read / Write

Default value: 16

Since: 3.13.2

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsOssDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsOssDevin

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.13.2

8.6 AgsOssDevout

AgsOssDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSS_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_oss_devout_error_quark ()
gboolean	ags_oss_devout_test_flags ()
void	ags_oss_devout_set_flags ()
void	ags_oss_devout_unset_flags ()
void	ags_oss_devout_switch_buffer_flag ()
void	ags_oss_devout_adjust_delay_and_attack ()
void	ags_oss_devout_realloc_buffer ()
AgsOssDevout *	ags_oss_devout_new ()
#define	AGS_IS_OSS_DEVOUT()
#define	AGS_IS_OSS_DEVOUT_CLASS()
#define	AGS_OSS_DEVOUT()
#define	AGS_OSS_DEVOUT_CLASS()
#define	AGS_OSS_DEVOUT_GET_CLASS()
GType	ags_oss_devout_flags_get_type ()
GType	ags_oss_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write

guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_OSS_DEVOUT_DEFAULT_OSS_DEVICE
#define	AGS_OSS_DEVOUT_DEFAULT_OSS_MIXER_DEVICE
#define	AGS_OSS_DEVOUT_DEFAULT_APP_BUFFER_SIZE
#define	AGS_OSS_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE
enum	AgsOssDevoutFlags
enum	AgsOssDevoutAppBufferMode
enum	AgsOssDevoutBackendBufferMode
#define	AGS_OSS_DEVOUT_ERROR
enum	AgsOssDevoutError
#define	AGS_TYPE_OSS_DEVOUT
struct	AgsOssDevout
struct	AgsOssDevoutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOssDevoutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsOssDevout
```

Implemented Interfaces

AgsOssDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/ags_oss_devout.h>
```

Description

AgsOssDevout represents a soundcard and supports output.

Functions

AGS_OSS_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_OSS_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsOssDevout *) obj)->obj_mutex))
```

ags_oss_devout_error_quark ()

```
GQuark
ags_oss_devout_error_quark ();
```

ags_oss_devout_test_flags ()

```
gboolean
ags_oss_devout_test_flags (AgsOssDevout *oss_devout,
                           guint flags);
```

Test *flags* to be set on *oss_devout* .

Parameters

oss_devout	the AgsOssDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.13.2

ags_oss_devout_set_flags ()

```
void
ags_oss_devout_set_flags (AgsOssDevout *oss_devout,
                          guint flags);
```

Enable a feature of *oss_devout* .

Parameters

oss_devout	the AgsOssDevout	
flags	see AgsOssDevoutFlags	

Since: 3.13.2

ags_oss_devout_unset_flags ()

```
void
ags_oss_devout_unset_flags (AgsOssDevout *oss_devout,
                            guint flags);
```

Disable a feature of *oss_devout* .

Parameters

oss_devout	the AgsOssDevout	
flags	see AgsOssDevoutFlags	

Since: 3.13.2

ags_oss_devout_switch_buffer_flag ()

```
void
ags_oss_devout_switch_buffer_flag (AgsOssDevout *oss_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

oss_devout	the AgsOssDevout

Since: 3.13.2

ags_oss_devout_adjust_delay_and_attack ()

```
void
ags_oss_devout_adjust_delay_and_attack
                                (AgsOssDevout *oss_devout);
```

Calculate delay and attack and reset it.

Parameters

oss_devout	the AgsOssDevout

Since: 3.13.2

ags_oss_devout_realloc_buffer ()

```
void
ags_oss_devout_realloc_buffer (AgsOssDevout *oss_devout);
```

Reallocate the internal audio buffer.

Parameters

oss_devout	the AgsOssDevout

Since: 3.13.2

ags_oss_devout_new ()

```
AgsOssDevout~*
ags_oss_devout_new ();
```

Creates a new instance of AgsOssDevout.

Returns

the new AgsOssDevout

Since: 3.13.2

AGS_IS_OSS_DEVOUT()

```
#define AGS_IS_OSS_DEVOUT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSS_DEVOUT))
```

AGS_IS_OSS_DEVOUT_CLASS()

```
#define AGS_IS_OSS_DEVOUT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSS_DEVOUT))
```

AGS_OSS_DEVOUT()

```
#define AGS_OSS_DEVOUT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSS_DEVOUT, AgsOssDevout))
```

AGS_OSS_DEVOUT_CLASS()

```
#define AGS_OSS_DEVOUT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSS_DEVOUT, AgsOssDevout))
```

AGS_OSS_DEVOUT_GET_CLASS()

```
#define AGS_OSS_DEVOUT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSS_DEVOUT, AgsOssDevoutClass))
```

ags_oss_devout_flags_get_type ()

```
GType
ags_oss_devout_flags_get_type ();
```

ags_oss_devout_get_type ()

```
GType
ags_oss_devout_get_type ();
```

Types and Values

AGS_OSS_DEVOUT_DEFAULT_OSS_DEVICE

```
#define AGS_OSS_DEVOUT_DEFAULT_OSS_DEVICE "/dev/dsp"
```

AGS_OSS_DEVOUT_DEFAULT_OSS_MIXER_DEVICE

```
#define AGS_OSS_DEVOUT_DEFAULT_OSS_MIXER_DEVICE "/dev/mixer"
```

AGS_OSS_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_OSS_DEVOUT_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_OSS_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE

```
#define AGS_OSS_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE (8)
```

enum AgsOssDevoutFlags

Enum values to control the behavior or indicate internal state of AgsOssDevout by enable/disable as flags.

Members

AGS_OSS_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_OSS_DEVOUT_PLAY	devout is run-ning
AGS_OSS_DEVOUT_START_PLAY	playback start-ing
AGS_OSS_DEVOUT_SHUTDOWN	stop play-back
AGS_OSS_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_OSS_DEVOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsOssDevoutAppBufferMode

Enum values to indicate internal state of AgsOssDevout application buffer by setting mode.

Members

AGS_OSS_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_OSS_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_OSS_DEVOUT_APP_BUFFER_2	ring-buffer 2
AGS_OSS_DEVOUT_APP_BUFFER_3	ring-buffer 3

enum AgsOssDevoutBackendBufferMode

Enum values to indicate internal state of AgsOssDevout backend buffer by setting mode.

Members

AGS_OSS_DEVOUT_BACKEND_BUFFER_0	ring-buffer 0
AGS_OSS_DEVOUT_BACKEND_BUFFER_1	ring-buffer 1
AGS_OSS_DEVOUT_BACKEND_BUFFER_2	ring-buffer 2
AGS_OSS_DEVOUT_BACKEND_BUFFER_3	ring-buffer 3
AGS_OSS_DEVOUT_BACKEND_BUFFER_4	ring-buffer 4
AGS_OSS_DEVOUT_BACKEND_BUFFER_5	ring-buffer 5
AGS_OSS_DEVOUT_BACKEND_BUFFER_6	ring-buffer 6
AGS_OSS_DEVOUT_BACKEND_BUFFER_7	ring-buffer 7

AGS_OSS_DEVOUT_ERROR

```
#define AGS_OSS_DEVOUT_ERROR (ags_oss_devout_error_quark())
```

enum AgsOssDevoutError

Members

AGS_OSS_DEVOUT_ERROR_LOCKED_SOUNDCARD		
AGS_OSS_DEVOUT_ERROR_BROKEN_CONFIGURATION		
AGS_OSS_DEVOUT_ERROR_ACCESS_TYPE_NOT_AVAILABLE		
AGS_OSS_DEVOUT_ERROR_SAMPLE_FORMAT_NOT_AVAILABLE		
AGS_OSS_DEVOUT_ERROR_CHANNELS_NOT_AVAILABLE		
AGS_OSS_DEVOUT_ERROR_SAMPLERATE_NOT_AVAILABLE		
AGS_OSS_DEVOUT_ERROR_BUFFER_SIZE_NOT_AVAILABLE		
AGS_OSS_DEVOUT_ERROR_HW_PARAMETERS_NOT_AVAILABLE		

AGS_TYPE_OSS_DEVOUT

```
#define AGS_TYPE_OSS_DEVOUT                (ags_oss_devout_get_type())
```

struct AgsOssDevout

```
struct AgsOssDevout;
```

struct AgsOssDevoutClass

```
struct AgsOssDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsOssDevout

Flags: Read

Since: 3.13.2

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsOssDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.13.2

The "buffer" property

```
"buffer"                    gpointer
```

The buffer

Owner: AgsOssDevout

Flags: Read

Since: 3.13.2

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size

Owner: AgsOssDevout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.13.2

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsOssDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.13.2

The "device" property

```
"device"                    char~*
```

The OSS soundcard indentifier

Owner: AgsOssDevout

Flags: Read / Write

Default value: "/dev/dsp"

Since: 3.13.2

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsOssDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsOssDevout

Flags: Read / Write

Default value: 16

Since: 3.13.2

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsOssDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.13.2

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsOssDevout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.13.2

8.7 AgsOssMidiin

AgsOssMidiin — Input from sequencer

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSS_MIDIIN_GET_OBJ_MUTEX()
GQuark	ags_oss_midiin_error_quark ()
gboolean	ags_oss_midiin_test_flags ()
void	ags_oss_midiin_set_flags ()
void	ags_oss_midiin_unset_flags ()
void	ags_oss_midiin_switch_buffer_flag ()
AgsOssMidiin *	ags_oss_midiin_new ()
#define	AGS_IS_OSS_MIDIIN()
#define	AGS_IS_OSS_MIDIIN_CLASS()
#define	AGS_OSS_MIDIIN()
#define	AGS_OSS_MIDIIN_CLASS()
#define	AGS_OSS_MIDIIN_GET_CLASS()
GType	ags_oss_midiin_flags_get_type ()
GType	ags_oss_midiin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
double	delay-factor	Read / Write
char *	device	Read / Write

Types and Values

#define	AGS_OSS_MIDIIN_DEFAULT_OSS_DEVICE
#define	AGS_OSS_MIDIIN_DEFAULT_OSS_MIXER_DEVICE
#define	AGS_OSS_MIDIIN_DEFAULT_APP_BUFFER_SIZE
#define	AGS_OSS_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE
#define	AGS_OSS_MIDIIN_DEFAULT_BUFFER_SIZE
enum	AgsOssMidiinFlags
enum	AgsOssMidiinAppBufferMode
enum	AgsOssMidiinBackendBufferMode
enum	AgsOssMidiinSyncFlags
#define	AGS_OSS_MIDIIN_ERROR
enum	AgsOssMidiinError
#define	AGS_TYPE_OSS_MIDIIN
struct	AgsOssMidiin
struct	AgsOssMidiinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOssMidiinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsOssMidiin
```

Implemented Interfaces

AgsOssMidiin implements AgsConnectable and AgsSequencer.

Includes

```
#include <ags/audio/ags_oss_midiin.h>
```

Description

AgsOssMidiin represents a sequencer and supports midi input.

Functions

AGS_OSS_MIDIIN_GET_OBJ_MUTEX()

```
#define AGS_OSS_MIDIIN_GET_OBJ_MUTEX(obj) (&(((AgsOssMidiin *) obj)->obj_mutex))
```

ags_oss_midiin_error_quark ()

```
GQuark
ags_oss_midiin_error_quark ();
```

ags_oss_midiin_test_flags ()

```
gboolean
ags_oss_midiin_test_flags (AgsOssMidiin *oss_midiin,
                           guint flags);
```

Test *flags* to be set on *oss_midiin* .

Parameters

oss_midiin	the AgsOssMidiin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.13.2

ags_oss_midiin_set_flags ()

```
void
ags_oss_midiin_set_flags (AgsOssMidiin *oss_midiin,
                          guint flags);
```

Enable a feature of *oss_midiin* .

Parameters

oss_midiin	the AgsOssMidiin	
flags	see AgsOssMidiinFlags	

Since: 3.13.2

ags_oss_midiin_unset_flags ()

```
void
ags_oss_midiin_unset_flags (AgsOssMidiin *oss_midiin,
                            guint flags);
```

Disable a feature of *oss_midiin* .

Parameters

oss_midiin	the AgsOssMidiin	
flags	see AgsOssMidiinFlags	

Since: 3.13.2

ags_oss_midiin_switch_buffer_flag ()

```
void
ags_oss_midiin_switch_buffer_flag (AgsOssMidiin *oss_midiin);
```

The buffer flag indicates the currently played buffer.

Parameters

oss_midiin	the AgsOssMidiin	

Since: 3.13.2

ags_oss_midiin_new ()

```
AgsOssMidiin~*
ags_oss_midiin_new ();
```

Creates a new instance of AgsOssMidiin.

Returns

the new AgsOssMidiin

Since: 3.13.2

AGS_IS_OSS_MIDIIN()

```
#define AGS_IS_OSS_MIDIIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSS_MIDIIN))
```

AGS_IS_OSS_MIDIIN_CLASS()

```
#define AGS_IS_OSS_MIDIIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSS_MIDIIN))
```

AGS_OSS_MIDIIN()

```
#define AGS_OSS_MIDIIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSS_MIDIIN, AgsOssMidiin))
```

AGS_OSS_MIDIIN_CLASS()

```
#define AGS_OSS_MIDIIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSS_MIDIIN, AgsOssMidiin))
```

AGS_OSS_MIDIIN_GET_CLASS()

```
#define AGS_OSS_MIDIIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSS_MIDIIN, AgsOssMidiinClass))
```

ags_oss_midiin_flags_get_type ()

```
GType
ags_oss_midiin_flags_get_type ();
```

ags_oss_midiin_get_type ()

```
GType
ags_oss_midiin_get_type ();
```

Types and Values

AGS_OSS_MIDIIN_DEFAULT_OSS_DEVICE

```
#define AGS_OSS_MIDIIN_DEFAULT_OSS_DEVICE "/dev/midi00"
```

AGS_OSS_MIDIIN_DEFAULT_OSS_MIXER_DEVICE

```
#define AGS_OSS_MIDIIN_DEFAULT_OSS_MIXER_DEVICE "/dev/mixer"
```

AGS_OSS_MIDIIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_OSS_MIDIIN_DEFAULT_APP_BUFFER_SIZE (4)
```

AGS_OSS_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE

```
#define AGS_OSS_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE (8)
```

AGS_OSS_MIDIIN_DEFAULT_BUFFER_SIZE

```
#define AGS_OSS_MIDIIN_DEFAULT_BUFFER_SIZE (4096)
```

enum AgsOssMidiinFlags

Enum values to control the behavior or indicate internal state of AgsOssMidiin by enable/disable as flags.

Members

AGS_OSS_MIDIIN_INITIALIZED	recording is initialized
AGS_OSS_MIDIIN_START_RECORD	just started recording
AGS_OSS_MIDIIN_RECORD	is recording
AGS_OSS_MIDIIN_SHUTDOWN	stop recording
AGS_OSS_MIDIIN_NONBLOCKING	do non-blocking calls

AGS_OSS_MIDIIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsOssMidiinAppBufferMode

Enum values to indicate internal state of AgsOssMidiin application buffer by setting mode.

Members

AGS_OSS_MIDIIN_APP_BUFFER_0	ring-buffer 0
AGS_OSS_MIDIIN_APP_BUFFER_1	ring-buffer 1
AGS_OSS_MIDIIN_APP_BUFFER_2	ring-buffer 2
AGS_OSS_MIDIIN_APP_BUFFER_3	ring-buffer 3

enum AgsOssMidiinBackendBufferMode

Enum values to indicate internal state of AgsOssMidiin backend buffer by setting mode.

Members

AGS_OSS_MIDIIN_BACKEND_BUFFER_0	ring-buffer 0
AGS_OSS_MIDIIN_BACKEND_BUFFER_1	ring-buffer 1
AGS_OSS_MIDIIN_BACKEND_BUFFER_2	ring-buffer 2
AGS_OSS_MIDIIN_BACKEND_BUFFER_3	ring-buffer 3
AGS_OSS_MIDIIN_BACKEND_BUFFER_4	ring-buffer 4

AGS_OSS_MIDIIN_BACKEND_BUFFER_5	ring-buffer 5
AGS_OSS_MIDIIN_BACKEND_BUFFER_6	ring-buffer 6
AGS_OSS_MIDIIN_BACKEND_BUFFER_7	ring-buffer 7

enum AgsOssMidiinSyncFlags

Enum values to control the synchronization between sequencer and poll.

Members

AGS_OSS_MIDIIN_PASS_THROUGH	do not sync
AGS_OSS_MIDIIN_INITIAL_POLL	initial poll
AGS_OSS_MIDIIN_POLL_WAIT	sync wait, sequencer conditional lock
AGS_OSS_MIDIIN_POLL_DONE	sync done, sequencer conditional lock
AGS_OSS_MIDIIN_POLL_FINISH_WAIT	sync wait, client conditional lock
AGS_OSS_MIDIIN_POLL_FINISH_DONE	sync done, client conditional lock
AGS_OSS_MIDIIN_POLL_SWITCH_BUFFER	switch buffer

AGS_OSS_MIDIIN_ERROR

```
#define AGS_OSS_MIDIIN_ERROR (ags_oss_midiin_error_quark())
```

enum AgsOssMidiinError

Members

AGS_OSS_MIDIIN_ERROR_LOCKED_SEQUENCER	

AGS_TYPE_OSS_MIDIIN

```
#define AGS_TYPE_OSS_MIDIIN                (ags_oss_midiin_get_type())
```

struct AgsOssMidiin

```
struct AgsOssMidiin;
```

struct AgsOssMidiinClass

```
struct AgsOssMidiinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsOssMidiin

Flags: Read

Since: 3.13.2

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsOssMidiin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.13.2

The "buffer" property

```
"buffer"                    gpointer
```

The buffer

Owner: AgsOssMidiin

Flags: Read

Since: 3.13.2

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsOssMidiin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.13.2

The "device" property

```
"device"                    char~*
```

The OSS sequencer indentifier

Owner: AgsOssMidiin

Flags: Read / Write

Default value: "/dev/midi00"

Since: 3.13.2

8.8 AgsWasapiDevout

AgsWasapiDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_WASAPI_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_wasapi_devout_error_quark ()
gboolean	ags_wasapi_devout_test_flags ()
void	ags_wasapi_devout_set_flags ()

void	ags_wasapi_devout_unset_flags ()
void	ags_wasapi_devout_switch_buffer_flag ()
void	ags_wasapi_devout_adjust_delay_and_attack ()
void	ags_wasapi_devout_realloc_buffer ()
AgsWasapiDevout *	ags_wasapi_devout_new ()
#define	AGS_IS_WASAPI_DEVOUT()
#define	AGS_IS_WASAPI_DEVOUT_CLASS()
#define	AGS_WASAPI_DEVOUT()
#define	AGS_WASAPI_DEVOUT_CLASS()
#define	AGS_WASAPI_DEVOUT_GET_CLASS()
GType	ags_wasapi_devout_flags_get_type ()
GType	ags_wasapi_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_WASAPI_DEVOUT_DEFAULT_APP_BUFFER_SIZE
#define	AGS_WASAPI_DEVOUT_DEFAULT_WASAPI_BUFFER_SIZE
enum	AgsWasapiDevoutFlags
enum	AgsWasapiDevoutAppBufferMode
#define	AGS_WASAPI_DEVOUT_ERROR
enum	AgsWasapiDevoutError
#define	AGS_TYPE_WASAPI_DEVOUT
struct	AgsWasapiDevout
struct	AgsWasapiDevoutClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsWasapiDevoutFlags
GObject
&#x2570;&#x2500;&#x2500; AgsWasapiDevout
```

Implemented Interfaces

AgsWasapiDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/wasapi/ags_wasapi_devout.h>
```

Description

AgsWasapiDevout represents a soundcard and supports output.

Functions

AGS_WASAPI_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_WASAPI_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsWasapiDevout *) obj)->obj_mutex))
```

ags_wasapi_devout_error_quark ()

```
GQuark
ags_wasapi_devout_error_quark ();
```

ags_wasapi_devout_test_flags ()

```
gboolean
ags_wasapi_devout_test_flags (AgsWasapiDevout *wasapi_devout,
                              guint flags);
```

Test *flags* to be set on *wasapi_devout* .

Parameters

wasapi_devout	the AgsWasapiDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_wasapi_devout_set_flags ()

```
void
ags_wasapi_devout_set_flags (AgsWasapiDevout *wasapi_devout,
                             guint flags);
```

Enable a feature of *wasapi_devout* .

Parameters

wasapi_devout	the AgsWasapiDevout	
flags	see AgsWasapiDevoutFlags	

Since: 3.0.0

ags_wasapi_devout_unset_flags ()

```
void
ags_wasapi_devout_unset_flags (AgsWasapiDevout *wasapi_devout,
                               guint flags);
```

Disable a feature of *wasapi_devout* .

Parameters

wasapi_devout	the AgsWasapiDevout	
flags	see AgsWasapiDevoutFlags	

Since: 3.0.0

ags_wasapi_devout_switch_buffer_flag ()

```
void
ags_wasapi_devout_switch_buffer_flag (AgsWasapiDevout *wasapi_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

wasapi_devout	an AgsWasapiDevout	

Since: 3.0.0

ags_wasapi_devout_adjust_delay_and_attack ()

```
void
ags_wasapi_devout_adjust_delay_and_attack
                               (AgsWasapiDevout *wasapi_devout);
```

Calculate delay and attack and reset it.

Parameters

wasapi_devout	the AgsWasapiDevout	

Since: 3.0.0

ags_wasapi_devout_realloc_buffer ()

```
void
ags_wasapi_devout_realloc_buffer (AgsWasapiDevout *wasapi_devout);
```

Reallocate the internal audio buffer.

Parameters

wasapi_devout	the AgsWasapiDevout

Since: 3.0.0

ags_wasapi_devout_new ()

```
AgsWasapiDevout~*
ags_wasapi_devout_new ();
```

Creates a new instance of AgsWasapiDevout.

Returns

a new AgsWasapiDevout

Since: 3.0.0

AGS_IS_WASAPI_DEVOUT()

```
#define AGS_IS_WASAPI_DEVOUT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_WASAPI_DEVOUT))
```

AGS_IS_WASAPI_DEVOUT_CLASS()

```
#define AGS_IS_WASAPI_DEVOUT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_WASAPI_DEVOUT))
```

AGS_WASAPI_DEVOUT()

```
#define AGS_WASAPI_DEVOUT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_WASAPI_DEVOUT, AgsWasapiDevout))
```

AGS_WASAPI_DEVOUT_CLASS()

```
#define AGS_WASAPI_DEVOUT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_WASAPI_DEVOUT, AgsWasapiDevout))
```

AGS_WASAPI_DEVOUT_GET_CLASS()

```
#define AGS_WASAPI_DEVOUT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_WASAPI_DEVOUT, AgsWasapiDevoutClass))
```

ags_wasapi_devout_flags_get_type ()

```
GType
ags_wasapi_devout_flags_get_type ();
```

ags_wasapi_devout_get_type ()

```
GType
ags_wasapi_devout_get_type ();
```

Types and Values

AGS_WASAPI_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_WASAPI_DEVOUT_DEFAULT_APP_BUFFER_SIZE (8)
```

AGS_WASAPI_DEVOUT_DEFAULT_WASAPI_BUFFER_SIZE

```
#define AGS_WASAPI_DEVOUT_DEFAULT_WASAPI_BUFFER_SIZE (8192)
```

enum AgsWasapiDevoutFlags

Enum values to control the behavior or indicate internal state of AgsWasapiDevout by enable/disable as flags.

Members

AGS_WASAPI_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_WASAPI_DEVOUT_START_PLAY	playback start-ing
AGS_WASAPI_DEVOUT_PLAY	do play-back
AGS_WASAPI_DEVOUT_SHUTDOWN	stop play-back
AGS_WASAPI_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_WASAPI_DEVOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

AGS_WASAPI_DEVOUT_SHARE_MODE_EXCLUSIVE	share mode exclusive

enum AgsWasapiDevoutAppBufferMode

Enum values to indicate internal state of AgsWasapiDevout application buffer by setting mode.

Members

AGS_WASAPI_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_WASAPI_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_WASAPI_DEVOUT_APP_BUFFER_2	ring-buffer 2
AGS_WASAPI_DEVOUT_APP_BUFFER_3	ring-buffer 3
AGS_WASAPI_DEVOUT_APP_BUFFER_4	ring-buffer 4
AGS_WASAPI_DEVOUT_APP_BUFFER_5	ring-buffer 5
AGS_WASAPI_DEVOUT_APP_BUFFER_6	ring-buffer 6
AGS_WASAPI_DEVOUT_APP_BUFFER_7	ring-buffer 7

AGS_WASAPI_DEVOUT_ERROR

```
#define AGS_WASAPI_DEVOUT_ERROR (ags_wasapi_devout_error_quark())
```

enum AgsWasapiDevoutError

Members

AGS_WASAPI_DEVOUT_ERROR_LOCKED_SOUNDCARD	
AGS_WASAPI_DEVOUT_ERROR_BROKEN_CONFIGURATION	

AGS_TYPE_WASAPI_DEVOUT

```
#define AGS_TYPE_WASAPI_DEVOUT                (ags_wasapi_devout_get_type())
```

struct AgsWasapiDevout

```
struct AgsWasapiDevout;
```

struct AgsWasapiDevoutClass

```
struct AgsWasapiDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsWasapiDevout

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsWasapiDevout

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [1,44100]

Default value: 940

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"               double
```

tact

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                     char~*
```

The core audio soundcard indentifier

Owner: AgsWasapiDevout

Flags: Read / Write

Default value: "ags-wasapi-devout-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"               guint
```

The dsp channel count

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsWasapiDevout

Flags: Read / Write

Allowed values: [8000,96000]

Default value: 44100

Since: 3.0.0

8.9 AgsWasapiDevin

AgsWasapiDevin — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_WASAPI_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_wasapi_devin_error_quark ()
gboolean	ags_wasapi_devin_test_flags ()
void	ags_wasapi_devin_set_flags ()
void	ags_wasapi_devin_unset_flags ()
void	ags_wasapi_devin_switch_buffer_flag ()
void	ags_wasapi_devin_adjust_delay_and_attack ()
void	ags_wasapi_devin_realloc_buffer ()
AgsWasapiDevin *	ags_wasapi_devin_new ()
#define	AGS_IS_WASAPI_DEVIN()
#define	AGS_IS_WASAPI_DEVIN_CLASS()
#define	AGS_WASAPI_DEVIN()
#define	AGS_WASAPI_DEVIN_CLASS()
#define	AGS_WASAPI_DEVIN_GET_CLASS()
GType	ags_wasapi_devin_flags_get_type ()
GType	ags_wasapi_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_WASAPI_DEVIN_DEFAULT_APP_BUFFER_SIZE
#define	AGS_WASAPI_DEVIN_DEFAULT_WASAPI_BUFFER_SIZE
enum	AgsWasapiDevinFlags
enum	AgsWasapiDevinAppBufferMode
#define	AGS_WASAPI_DEVIN_ERROR
enum	AgsWasapiDevinError
#define	AGS_TYPE_WASAPI_DEVIN
struct	AgsWasapiDevin
struct	AgsWasapiDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsWasapiDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsWasapiDevin
```

Implemented Interfaces

AgsWasapiDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/wasapi/ags_wasapi_devin.h>
```

Description

AgsWasapiDevin represents a soundcard and supports output.

Functions

AGS_WASAPI_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_WASAPI_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsWasapiDevin *) obj)->obj_mutex))
```

ags_wasapi_devin_error_quark ()

```
GQuark
ags_wasapi_devin_error_quark ();
```

ags_wasapi_devin_test_flags ()

```
gboolean
ags_wasapi_devin_test_flags (AgsWasapiDevin *wasapi_devin,
                             guint flags);
```

Test *flags* to be set on *wasapi_devin* .

Parameters

wasapi_devin	the AgsWasapiDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_wasapi_devin_set_flags ()

```
void
ags_wasapi_devin_set_flags (AgsWasapiDevin *wasapi_devin,
                            guint flags);
```

Enable a feature of *wasapi_devin* .

Parameters

wasapi_devin	the AgsWasapiDevin	
flags	see AgsWasapiDevinFlags	

Since: 3.0.0

ags_wasapi_devin_unset_flags ()

```
void
ags_wasapi_devin_unset_flags (AgsWasapiDevin *wasapi_devin,
                              guint flags);
```

Disable a feature of *wasapi_devin*.

Parameters

wasapi_devin	the AgsWasapiDevin	
flags	see AgsWasapiDevinFlags	

Since: 3.0.0

ags_wasapi_devin_switch_buffer_flag ()

```
void
ags_wasapi_devin_switch_buffer_flag (AgsWasapiDevin *wasapi_devin);
```

The buffer flag indicates the currently played buffer.

Parameters

wasapi_devin	an AgsWasapiDevin	

Since: 3.0.0

ags_wasapi_devin_adjust_delay_and_attack ()

```
void
ags_wasapi_devin_adjust_delay_and_attack
                              (AgsWasapiDevin *wasapi_devin);
```

Calculate delay and attack and reset it.

Parameters

wasapi_devin	the AgsWasapiDevin	

Since: 3.0.0

ags_wasapi_devin_realloc_buffer ()

```
void
ags_wasapi_devin_realloc_buffer (AgsWasapiDevin *wasapi_devin);
```

Reallocate the internal audio buffer.

Parameters

wasapi_devin	the AgsWasapiDevin

Since: 3.0.0

ags_wasapi_devin_new ()

```
AgsWasapiDevin~*
ags_wasapi_devin_new ();
```

Creates a new instance of AgsWasapiDevin.

Returns

a new AgsWasapiDevin

Since: 3.0.0

AGS_IS_WASAPI_DEVIN()

```
#define AGS_IS_WASAPI_DEVIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_WASAPI_DEVIN))
```

AGS_IS_WASAPI_DEVIN_CLASS()

```
#define AGS_IS_WASAPI_DEVIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_WASAPI_DEVIN))
```

AGS_WASAPI_DEVIN()

```
#define AGS_WASAPI_DEVIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_WASAPI_DEVIN, AgsWasapiDevin))
```

AGS_WASAPI_DEVIN_CLASS()

```
#define AGS_WASAPI_DEVIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_WASAPI_DEVIN, AgsWasapiDevin))
```

AGS_WASAPI_DEVIN_GET_CLASS()

```
#define AGS_WASAPI_DEVIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_WASAPI_DEVIN, AgsWasapiDevinClass))
```

ags_wasapi_devin_flags_get_type ()

```
GType
ags_wasapi_devin_flags_get_type ();
```

ags_wasapi_devin_get_type ()

```
GType
ags_wasapi_devin_get_type ();
```

Types and Values

AGS_WASAPI_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_WASAPI_DEVIN_DEFAULT_APP_BUFFER_SIZE (8)
```

AGS_WASAPI_DEVIN_DEFAULT_WASAPI_BUFFER_SIZE

```
#define AGS_WASAPI_DEVIN_DEFAULT_WASAPI_BUFFER_SIZE (8192)
```

enum AgsWasapiDevinFlags

Enum values to control the behavior or indicate internal state of AgsWasapiDevin by enable/disable as flags.

Members

AGS_WASAPI_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_WASAPI_DEVIN_START_RECORD	capture start-ing
AGS_WASAPI_DEVIN_RECORD	do cap-ture
AGS_WASAPI_DEVIN_SHUTDOWN	stop cap-ture
AGS_WASAPI_DEVIN_NONBLOCKING	do non-blocking calls

AGS_WASAPI_DEVIN_ATTACK_FIRST	use first attack, instead of second one
AGS_WASAPI_DEVIN_SHARE_MODE_EXCLUSIVE	share mode exclusive

enum AgsWasapiDevinAppBufferMode

Enum values to indicate internal state of AgsWasapiDevin application buffer by setting mode.

Members

AGS_WASAPI_DEVIN_APP_BUFFER_0	ring-buffer 0
AGS_WASAPI_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_WASAPI_DEVIN_APP_BUFFER_2	ring-buffer 2
AGS_WASAPI_DEVIN_APP_BUFFER_3	ring-buffer 3
AGS_WASAPI_DEVIN_APP_BUFFER_4	ring-buffer 4
AGS_WASAPI_DEVIN_APP_BUFFER_5	ring-buffer 5
AGS_WASAPI_DEVIN_APP_BUFFER_6	ring-buffer 6
AGS_WASAPI_DEVIN_APP_BUFFER_7	ring-buffer 7

AGS_WASAPI_DEVIN_ERROR

```
#define AGS_WASAPI_DEVIN_ERROR (ags_wasapi_devin_error_quark())
```

enum AgsWasapiDevinError

Members

AGS_WASAPI_DEVIN_ERROR_LOCKED_SOUNDCARD	
AGS_WASAPI_DEVIN_ERROR_BROKEN_CONFIGURATION	

AGS_TYPE_WASAPI_DEVIN

```
#define AGS_TYPE_WASAPI_DEVIN                (ags_wasapi_devin_get_type())
```

struct AgsWasapiDevin

```
struct AgsWasapiDevin;
```

struct AgsWasapiDevinClass

```
struct AgsWasapiDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsWasapiDevin

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
"buffer"                    gpointer
```

The buffer

Owner: AgsWasapiDevin

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [1,44100]

Default value: 940

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The core audio soundcard indentifier

Owner: AgsWasapiDevin

Flags: Read / Write

Default value: "ags-wasapi-devin-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsWasapiDevin

Flags: Read / Write

Allowed values: [8000,96000]

Default value: 44100

Since: 3.0.0

8.10 AgsJackServer

AgsJackServer — JACK instance

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_JACK_SERVER_GET_OBJ_MUTEX()
gboolean	ags_jack_server_test_flags ()
void	ags_jack_server_set_flags ()
void	ags_jack_server_unset_flags ()
GList *	ags_jack_server_find_url ()
GObject *	ags_jack_server_find_client ()
GObject *	ags_jack_server_find_port ()
void	ags_jack_server_add_client ()
void	ags_jack_server_remove_client ()
void	ags_jack_server_connect_client ()
void	ags_jack_server_disconnect_client ()
AgsJackServer *	ags_jack_server_new ()
#define	AGS_IS_JACK_SERVER()
#define	AGS_IS_JACK_SERVER_CLASS()
#define	AGS_JACK_SERVER()
#define	AGS_JACK_SERVER_CLASS()
#define	AGS_JACK_SERVER_GET_CLASS()
GType	ags_jack_server_get_type ()

Properties

AgsJackClient *	default-jack-client	Read / Write
GObject *	default-soundcard	Read / Write
AgsJackClient *	input-jack-client	Read / Write
gpointer	jack-client	Read / Write
char *	url	Read / Write

Types and Values

#define	AGS_JACK_SERVER_DEFAULT_PROTOCOL
#define	AGS_JACK_SERVER_DEFAULT_HOST
#define	AGS_JACK_SERVER_DEFAULT_PORT
#define	AGS_TYPE_JACK_SERVER
struct	AgsJackServer
struct	AgsJackServerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsJackServer
```

Implemented Interfaces

AgsJackServer implements AgsConnectable and AgsSoundServer.

Includes

```
#include <ags/audio/jack/ags_jack_server.h>
```

Description

The AgsJackServer is an object to represent a running JACK instance.

Functions

AGS_JACK_SERVER_GET_OBJ_MUTEX()

```
#define AGS_JACK_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsJackServer *) obj)->obj_mutex))
```

ags_jack_server_test_flags ()

```
gboolean
ags_jack_server_test_flags (AgsJackServer *jack_server,
                            guint flags);
```

Test *flags* to be set on *jack_server* .

Parameters

jack_server	the AgsJackServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_jack_server_set_flags ()

```
void
ags_jack_server_set_flags (AgsJackServer *jack_server,
                           guint flags);
```

Enable a feature of *jack_server* .

Parameters

jack_server	the AgsJackServer	
flags	see AgsJackServerFlags	

Since: 3.0.0

ags_jack_server_unset_flags ()

```
void
ags_jack_server_unset_flags (AgsJackServer *jack_server,
                             guint flags);
```

Disable a feature of *jack_server* .

Parameters

jack_server	the AgsJackServer	
flags	see AgsJackServerFlags	

Since: 3.0.0

ags_jack_server_find_url ()

```
GList~*
ags_jack_server_find_url (GList *jack_server,
                          gchar *url);
```

Find AgsJackServer by url.

Parameters

jack_server	the GList containing AgsJackServer.	*[element-type AgsAudio.JackServer][transfer none]*
url	the url to find	

Returns

the GList containing a AgsJackServer matching *url* or NULL.

[element-type AgsAudio.JackServer][transfer none]

Since: 3.0.0

ags_jack_server_find_client ()

```
GObject~*
ags_jack_server_find_client (AgsJackServer *jack_server,
                             gchar *client_uuid);
```

Find AgsJackClient by uuid.

Parameters

jack_server	the AgsJackServer	
client_uuid	the uuid to find	

Returns

the AgsJackClient found or NULL.

[transfer none]

Since: 3.0.0

ags_jack_server_find_port ()

```
GObject~*
ags_jack_server_find_port (AgsJackServer *jack_server,
                           gchar *port_uuid);
```

Find AgsJackPort by uuid.

Parameters

jack_server	the AgsJackServer	
port_uuid	the uuid to find	

Returns

the AgsJackPort found or NULL.

[transfer none]

Since: 3.0.0

ags_jack_server_add_client ()

```
void
ags_jack_server_add_client (AgsJackServer *jack_server,
                            GObject *jack_client);
```

Add *jack_client* to *jack_server*

Parameters

jack_server	the AgsJackServer	
jack_client	the AgsJackClient to add	

Since: 3.0.0

ags_jack_server_remove_client ()

```
void
ags_jack_server_remove_client (AgsJackServer *jack_server,
                               GObject *jack_client);
```

Remove *jack_client* to *jack_server*

Parameters

jack_server	the AgsJackServer	
jack_client	the AgsJackClient to remove	

Since: 3.0.0

ags_jack_server_connect_client ()

```
void
ags_jack_server_connect_client (AgsJackServer *jack_server);
```

Connect all clients.

Parameters

jack_server	the AgsJackServer	

Since: 3.0.0

ags_jack_server_disconnect_client ()

```
void
ags_jack_server_disconnect_client (AgsJackServer *jack_server);
```

Connect all clients.

Parameters

jack_server	the AgsJackServer	

Since: 3.0.0

ags_jack_server_new ()

```
AgsJackServer~*
ags_jack_server_new (gchar *url);
```

Create a new instance of AgsJackServer.

Parameters

url	the URL as string	

Returns

the new AgsJackServer

Since: 3.0.0

AGS_IS_JACK_SERVER()

```
#define AGS_IS_JACK_SERVER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_JACK_SERVER))
```

AGS_IS_JACK_SERVER_CLASS()

```
#define AGS_IS_JACK_SERVER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_JACK_SERVER))
```

AGS_JACK_SERVER()

```
#define AGS_JACK_SERVER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_JACK_SERVER, AgsJackServer))
```

AGS_JACK_SERVER_CLASS()

```
#define AGS_JACK_SERVER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_JACK_SERVER, AgsJackServer))
```

AGS_JACK_SERVER_GET_CLASS()

```
#define AGS_JACK_SERVER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_JACK_SERVER, AgsJackServerClass))
```

ags_jack_server_get_type ()

```
GType
ags_jack_server_get_type ();
```

Types and Values

AGS_JACK_SERVER_DEFAULT_PROTOCOL

```
#define AGS_JACK_SERVER_DEFAULT_PROTOCOL "udp"
```

AGS_JACK_SERVER_DEFAULT_HOST

```
#define AGS_JACK_SERVER_DEFAULT_HOST "localhost"
```

AGS_JACK_SERVER_DEFAULT_PORT

```
#define AGS_JACK_SERVER_DEFAULT_PORT (3000)
```

AGS_TYPE_JACK_SERVER

```
#define AGS_TYPE_JACK_SERVER                (ags_jack_server_get_type())
```

struct AgsJackServer

```
struct AgsJackServer;
```

struct AgsJackServerClass

```
struct AgsJackServerClass {
  GObjectClass gobject;
};
```

Property Details

The "default-jack-client" property

```
  "default-jack-client"       AgsJackClient~*
```

The default jack client.

Owner: AgsJackServer

Flags: Read / Write

Since: 3.0.0

The "default-soundcard" property

```
  "default-soundcard"         GObject~*
```

The default soundcard.

Owner: AgsJackServer

Flags: Read / Write

Since: 3.0.0

The "input-jack-client" property

```
  "input-jack-client"         AgsJackClient~*
```

The input jack client.

Owner: AgsJackServer

Flags: Read / Write

Since: 3.0.0

The "jack-client" property

```
"jack-client"              gpointer
```

The jack client list.

[transfer full]

Owner: AgsJackServer

Flags: Read / Write

Since: 3.0.0

The "url" property

```
"url"                      char~*
```

The assigned URL.

Owner: AgsJackServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

8.11 AgsJackClient

AgsJackClient — JACK connection

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_JACK_CLIENT_GET_OBJ_MUTEX()
gboolean	ags_jack_client_test_flags ()
void	ags_jack_client_set_flags ()
void	ags_jack_client_unset_flags ()
GList *	ags_jack_client_find_uuid ()
GList *	ags_jack_client_find ()
void	ags_jack_client_open ()
void	ags_jack_client_close ()
void	ags_jack_client_add_device ()
void	ags_jack_client_remove_device ()
void	ags_jack_client_add_port ()
void	ags_jack_client_remove_port ()
void	ags_jack_client_activate ()
void	ags_jack_client_deactivate ()
AgsJackClient *	ags_jack_client_new ()
#define	AGS_IS_JACK_CLIENT()
#define	AGS_IS_JACK_CLIENT_CLASS()

#define	AGS_JACK_CLIENT()
#define	AGS_JACK_CLIENT_CLASS()
#define	AGS_JACK_CLIENT_GET_CLASS()
GType	ags_jack_client_flags_get_type ()
GType	ags_jack_client_get_type ()

Properties

char *	client-name	Read / Write
gpointer	device	Read / Write
AgsJackServer *	jack-server	Read / Write
gpointer	port	Read / Write

Types and Values

enum	AgsJackClientFlags
#define	AGS_TYPE_JACK_CLIENT
struct	AgsJackClient
struct	AgsJackClientClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsJackClientFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsJackClient
```

Implemented Interfaces

AgsJackClient implements AgsConnectable.

Includes

```
#include <ags/audio/jack/ags_jack_client.h>
```

Description

The AgsJackClient communicates with a JACK instance.

Functions

AGS_JACK_CLIENT_GET_OBJ_MUTEX()

```
#define AGS_JACK_CLIENT_GET_OBJ_MUTEX(obj) (&(((AgsJackClient *) obj)->obj_mutex))
```

ags_jack_client_test_flags ()

```
gboolean
ags_jack_client_test_flags (AgsJackClient *jack_client,
                            guint flags);
```

Test *flags* to be set on *jack_client* .

Parameters

jack_client	the AgsJackClient	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_jack_client_set_flags ()

```
void
ags_jack_client_set_flags (AgsJackClient *jack_client,
                           guint flags);
```

Enable a feature of *jack_client* .

Parameters

jack_client	the AgsJackClient	
flags	see AgsJackClientFlags	

Since: 3.0.0

ags_jack_client_unset_flags ()

```
void
ags_jack_client_unset_flags (AgsJackClient *jack_client,
                             guint flags);
```

Disable a feature of *jack_client* .

Parameters

jack_client	the AgsJackClient	
flags	see AgsJackClientFlags	

Since: 3.0.0

ags_jack_client_find_uuid ()

```
GList~*
ags_jack_client_find_uuid (GList *jack_client,
                           gchar *client_uuid);
```

Finds next match of *client_uuid* in *jack_client* .

Parameters

jack_client	the GList containing AgsJackClient.	*[element-type AgsAudio.JackClient][transfer none]*
client_uuid	the client uuid to find	

Returns

the matching GList or NULL.

[element-type AgsAudio.JackClient][transfer none]

Since: 3.0.0

ags_jack_client_find ()

```
GList~*
ags_jack_client_find (GList *jack_client,
                      gchar *client_name);
```

Finds next match of *client_name* in *jack_client* .

Parameters

jack_client	the GList containing AgsJackClient.	*[element-type AgsAudio.JackClient][transfer none]*
client_name	the client name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.JackClient][transfer none]

Since: 3.0.0

ags_jack_client_open ()

```
void
ags_jack_client_open (AgsJackClient *jack_client,
                      gchar *client_name);
```

Open the JACK client's connection and read uuid.

Parameters

jack_client	the AgsJackClient	
client_name	the client's name	

Since: 3.0.0

ags_jack_client_close ()

```
void
ags_jack_client_close (AgsJackClient *jack_client);
```

Close the JACK client's connection.

Parameters

jack_client	the AgsJackClient	

Since: 3.0.0

ags_jack_client_add_device ()

```
void
ags_jack_client_add_device (AgsJackClient *jack_client,
                            GObject *jack_device);
```

Add *jack_device* to *jack_client* .

Parameters

jack_client	the AgsJackClient	
jack_device	an AgsJackDevout, AgsJackDevin or AgsJackMidiin	

Since: 3.0.0

ags_jack_client_remove_device ()

```
void
ags_jack_client_remove_device (AgsJackClient *jack_client,
                               GObject *jack_device);
```

Remove *jack_device* from *jack_client* .

Parameters

jack_client	the AgsJackClient	
jack_device	an AgsJackDevout, AgsJackDevin or AgsJackMidiin	

Since: 3.0.0

ags_jack_client_add_port ()

```
void
ags_jack_client_add_port (AgsJackClient *jack_client,
                          GObject *jack_port);
```

Add *jack_port* to *jack_client* .

Parameters

jack_client	the AgsJackClient	
jack_port	an AgsJackPort	

Since: 3.0.0

ags_jack_client_remove_port ()

```
void
ags_jack_client_remove_port (AgsJackClient *jack_client,
                             GObject *jack_port);
```

Remove *jack_port* from *jack_client* .

Parameters

jack_client	the AgsJackClient	
jack_port	an AgsJackPort	

Since: 3.0.0

ags_jack_client_activate ()

```
void
ags_jack_client_activate (AgsJackClient *jack_client);
```

Activate client.

Parameters

jack_client	the AgsJackClient	

Since: 3.0.0

ags_jack_client_deactivate ()

```
void
ags_jack_client_deactivate (AgsJackClient *jack_client);
```

Deactivate client.

Parameters

jack_client	the AgsJackClient

Since: 3.0.0

ags_jack_client_new ()

```
AgsJackClient~*
ags_jack_client_new (GObject *jack_server);
```

Create a new instance of AgsJackClient.

Parameters

jack_server	the assigned AgsJackServer

Returns

the new AgsJackClient

Since: 3.0.0

AGS_IS_JACK_CLIENT()

```
#define AGS_IS_JACK_CLIENT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_JACK_CLIENT))
```

AGS_IS_JACK_CLIENT_CLASS()

```
#define AGS_IS_JACK_CLIENT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_JACK_CLIENT))
```

AGS_JACK_CLIENT()

```
#define AGS_JACK_CLIENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_JACK_CLIENT, AgsJackClient))
```

AGS_JACK_CLIENT_CLASS()

```
#define AGS_JACK_CLIENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_JACK_CLIENT, AgsJackClient))
```

AGS_JACK_CLIENT_GET_CLASS()

```
#define AGS_JACK_CLIENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_JACK_CLIENT, AgsJackClientClass))
```

ags_jack_client_flags_get_type ()

```
GType
ags_jack_client_flags_get_type ();
```

ags_jack_client_get_type ()

```
GType
ags_jack_client_get_type ();
```

Types and Values

enum AgsJackClientFlags

Enum values to control the behavior or indicate internal state of AgsJackClient by enable/disable as flags.

Members

AGS_JACK_CLIENT_ACTIVATED	the client was activated

AGS_TYPE_JACK_CLIENT

```
#define AGS_TYPE_JACK_CLIENT                (ags_jack_client_get_type())
```

struct AgsJackClient

```
struct AgsJackClient;
```

struct AgsJackClientClass

```
struct AgsJackClientClass {
  GObjectClass gobject;
};
```

Property Details

The "client-name" property

```
"client-name"              char~*
```

The JACK client name.

Owner: AgsJackClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "device" property

```
"device"                   gpointer
```

The assigned devices.

[transfer full]

Owner: AgsJackClient

Flags: Read / Write

Since: 3.0.0

The "jack-server" property

```
"jack-server"              AgsJackServer~*
```

The assigned AgsJackServer.

Owner: AgsJackClient

Flags: Read / Write

Since: 3.0.0

The "port" property

```
"port"                     gpointer
```

The assigned ports.

[transfer full]

Owner: AgsJackClient

Flags: Read / Write

Since: 3.0.0

8.12 AgsJackPort

AgsJackPort — JACK resource.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_JACK_PORT_GET_OBJ_MUTEX()
gboolean	ags_jack_port_test_flags ()
void	ags_jack_port_set_flags ()
void	ags_jack_port_unset_flags ()
GList *	ags_jack_port_find ()
void	ags_jack_port_register ()
void	ags_jack_port_unregister ()
AgsJackPort *	ags_jack_port_new ()
#define	AGS_IS_JACK_PORT()
#define	AGS_IS_JACK_PORT_CLASS()
#define	AGS_JACK_PORT()
#define	AGS_JACK_PORT_CLASS()
#define	AGS_JACK_PORT_GET_CLASS()
GType	ags_jack_port_flags_get_type ()
GType	ags_jack_port_get_type ()

Properties

AgsJackClient *	jack-client	Read / Write
char *	port-name	Read / Write

Types and Values

enum	AgsJackPortFlags
#define	AGS_TYPE_JACK_PORT
struct	AgsJackPort
struct	AgsJackPortClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsJackPortFlags
GObject
&#x2570;&#x2500;&#x2500; AgsJackPort
```

Implemented Interfaces

AgsJackPort implements AgsConnectable.

Includes

```
#include <ags/audio/jack/ags_jack_port.h>
```

Description

The AgsJackPort represents either a JACK sequencer or soundcard to communicate with.

Functions

AGS_JACK_PORT_GET_OBJ_MUTEX()

```
#define AGS_JACK_PORT_GET_OBJ_MUTEX(obj) (&(((AgsJackPort *) obj)->obj_mutex))
```

ags_jack_port_test_flags ()

```
gboolean
ags_jack_port_test_flags (AgsJackPort *jack_port,
                          guint flags);
```

Test *flags* to be set on *jack_port* .

Parameters

jack_port	the AgsJackPort	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_jack_port_set_flags ()

```
void
ags_jack_port_set_flags (AgsJackPort *jack_port,
                         guint flags);
```

Enable a feature of *jack_port* .

Parameters

jack_port	the AgsJackPort	
flags	see AgsJackPortFlags	

Since: 3.0.0

ags_jack_port_unset_flags ()

```
void
ags_jack_port_unset_flags (AgsJackPort *jack_port,
                           guint flags);
```

Disable a feature of *jack_port* .

Parameters

jack_port	the AgsJackPort	
flags	see AgsJackPortFlags	

Since: 3.0.0

ags_jack_port_find ()

```
GList~*
ags_jack_port_find (GList *jack_port,
                    gchar *port_name);
```

Finds next match of *port_name* in *jack_port* .

Parameters

jack_port	the GList containing AgsJackPort.	*[element-type AgsAudio.JackPort][transfer none]*
port_name	the port name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.JackPort][transfer none]

Since: 3.0.0

ags_jack_port_register ()

```
void
ags_jack_port_register (AgsJackPort *jack_port,
                        gchar *port_name,
                        gboolean is_audio,
                        gboolean is_midi,
                        gboolean is_output);
```

Register a new JACK port and read uuid. Creates a new AgsSequencer or AgsSoundcard object.

Parameters

jack_port	the AgsJackPort	
port_name	the name as string	
is_audio	if TRUE interpreted as audio port	
is_midi	if TRUE interpreted as midi port	
is_output	if TRUE port is acting as output, otherwise as input	

Since: 3.0.0

ags_jack_port_unregister ()

```
void
ags_jack_port_unregister (AgsJackPort *jack_port);
```

Unregister JACK port.

Parameters

jack_port	the AgsJackPort

Since: 3.0.0

ags_jack_port_new ()

```
AgsJackPort~*
ags_jack_port_new (GObject *jack_client);
```

Create a new instance of AgsJackPort.

Parameters

jack_client	the AgsJackClient assigned to

Returns

the new AgsJackPort

Since: 3.0.0

AGS_IS_JACK_PORT()

```
#define AGS_IS_JACK_PORT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_JACK_PORT))
```

AGS_IS_JACK_PORT_CLASS()

```
#define AGS_IS_JACK_PORT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_JACK_PORT))
```

AGS_JACK_PORT()

```
#define AGS_JACK_PORT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_JACK_PORT, AgsJackPort))
```

AGS_JACK_PORT_CLASS()

```
#define AGS_JACK_PORT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_JACK_PORT, AgsJackPort))
```

AGS_JACK_PORT_GET_CLASS()

```
#define AGS_JACK_PORT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_JACK_PORT, AgsJackPortClass))
```

ags_jack_port_flags_get_type ()

```
GType
ags_jack_port_flags_get_type ();
```

ags_jack_port_get_type ()

```
GType
ags_jack_port_get_type ();
```

Types and Values

enum AgsJackPortFlags

Enum values to control the behavior or indicate internal state of AgsJackPort by enable/disable as flags.

Members

AGS_JACK_PORT_REGISTERED	the port was registered
AGS_JACK_PORT_IS_AUDIO	the port provides audio data
AGS_JACK_PORT_IS_MIDI	the port provides midi data
AGS_JACK_PORT_IS_OUTPUT	the port does output

AGS_JACK_PORT_IS_INPUT	the port does input

AGS_TYPE_JACK_PORT

```
#define AGS_TYPE_JACK_PORT                (ags_jack_port_get_type())
```

struct AgsJackPort

```
struct AgsJackPort;
```

struct AgsJackPortClass

```
struct AgsJackPortClass {
  GObjectClass gobject;
};
```

Property Details

The "jack-client" property

```
  "jack-client"              AgsJackClient~*
```

The assigned AgsJackClient.

Owner: AgsJackPort

Flags: Read / Write

Since: 3.0.0

The "port-name" property

```
  "port-name"                char~*
```

The JACK port name.

Owner: AgsJackPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

8.13 AgsJackDevin

AgsJackDevin — Input from soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_JACK_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_jack_devin_error_quark ()
gboolean	ags_jack_devin_test_flags ()
void	ags_jack_devin_set_flags ()
void	ags_jack_devin_unset_flags ()
void	ags_jack_devin_switch_buffer_flag ()
void	ags_jack_devin_adjust_delay_and_attack ()
void	ags_jack_devin_realloc_buffer ()
AgsJackDevin *	ags_jack_devin_new ()
#define	AGS_IS_JACK_DEVIN()
#define	AGS_IS_JACK_DEVIN_CLASS()
#define	AGS_JACK_DEVIN()
#define	AGS_JACK_DEVIN_CLASS()
#define	AGS_JACK_DEVIN_GET_CLASS()
GType	ags_jack_devin_flags_get_type ()
GType	ags_jack_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
AgsJackClient *	jack-client	Read / Write
gpointer	jack-port	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

enum	AgsJackDevinFlags
enum	AgsJackDevinAppBufferMode
enum	AgsJackDevinSyncFlags
#define	AGS_JACK_DEVIN_ERROR
enum	AgsJackDevinError
#define	AGS_TYPE_JACK_DEVIN
struct	AgsJackDevin
struct	AgsJackDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsJackDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsJackDevin
```

Implemented Interfaces

AgsJackDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/jack/ags_jack_devin.h>
```

Description

AgsJackDevin represents a soundcard and supports input.

Functions

AGS_JACK_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_JACK_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsJackDevin *) obj)->obj_mutex))
```

ags_jack_devin_error_quark ()

```
GQuark
ags_jack_devin_error_quark ();
```

ags_jack_devin_test_flags ()

```
gboolean
ags_jack_devin_test_flags (AgsJackDevin *jack_devin,
                           guint flags);
```

Test *flags* to be set on *jack_devin* .

Parameters

jack_devin	the AgsJackDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_jack_devin_set_flags ()

```
void
ags_jack_devin_set_flags (AgsJackDevin *jack_devin,
                          guint flags);
```

Enable a feature of *jack_devin* .

Parameters

jack_devin	the AgsJackDevin	
flags	see AgsJackDevinFlags	

Since: 3.0.0

ags_jack_devin_unset_flags ()

```
void
ags_jack_devin_unset_flags (AgsJackDevin *jack_devin,
                            guint flags);
```

Disable a feature of *jack_devin* .

Parameters

jack_devin	the AgsJackDevin	
flags	see AgsJackDevinFlags	

Since: 3.0.0

ags_jack_devin_switch_buffer_flag ()

```
void
ags_jack_devin_switch_buffer_flag (AgsJackDevin *jack_devin);
```

The buffer flag indicates the currently played buffer.

Parameters

jack_devin	an AgsJackDevin	

Since: 3.0.0

ags_jack_devin_adjust_delay_and_attack ()

```
void
ags_jack_devin_adjust_delay_and_attack
                                (AgsJackDevin *jack_devin);
```

Calculate delay and attack and reset it.

Parameters

jack_devin	the AgsJackDevin

Since: 3.0.0

ags_jack_devin_realloc_buffer ()

```
void
ags_jack_devin_realloc_buffer (AgsJackDevin *jack_devin);
```

Reallocate the internal audio buffer.

Parameters

jack_devin	the AgsJackDevin

Since: 3.0.0

ags_jack_devin_new ()

```
AgsJackDevin~*
ags_jack_devin_new ();
```

Creates a new instance of AgsJackDevin.

Returns

the new AgsJackDevin

Since: 3.0.0

AGS_IS_JACK_DEVIN()

```
#define AGS_IS_JACK_DEVIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_JACK_DEVIN))
```

AGS_IS_JACK_DEVIN_CLASS()

```
#define AGS_IS_JACK_DEVIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_JACK_DEVIN))
```

AGS_JACK_DEVIN()

```
#define AGS_JACK_DEVIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_JACK_DEVIN, AgsJackDevin))
```

AGS_JACK_DEVIN_CLASS()

```
#define AGS_JACK_DEVIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_JACK_DEVIN, AgsJackDevin))
```

AGS_JACK_DEVIN_GET_CLASS()

```
#define AGS_JACK_DEVIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_JACK_DEVIN, AgsJackDevinClass))
```

ags_jack_devin_flags_get_type ()

```
GType
ags_jack_devin_flags_get_type ();
```

ags_jack_devin_get_type ()

```
GType
ags_jack_devin_get_type ();
```

Types and Values

enum AgsJackDevinFlags

Enum values to control the behavior or indicate internal state of AgsJackDevin by enable/disable as flags.

Members

AGS_JACK_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_JACK_DEVIN_START_RECORD	capture start-ing
AGS_JACK_DEVIN_RECORD	do cap-ture
AGS_JACK_DEVIN_SHUTDOWN	stop cap-ture
AGS_JACK_DEVIN_NONBLOCKING	do non-blocking calls

AGS_JACK_DEVIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsJackDevinAppBufferMode

Enum values to indicate internal state of AgsJackDevin application buffer by setting mode.

Members

AGS_JACK_DEVIN_APP_BUFFER_0	ring-buffer 0
AGS_JACK_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_JACK_DEVIN_APP_BUFFER_2	ring-buffer 2
AGS_JACK_DEVIN_APP_BUFFER_3	ring-buffer 3

enum AgsJackDevinSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_JACK_DEVIN_PASS_THROUGH	do not sync
AGS_JACK_DEVIN_INITIAL_CALLBACK	initial call-back
AGS_JACK_DEVIN_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock

AGS_JACK_DEVIN_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_JACK_DEVIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_JACK_DEVIN_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_JACK_DEVIN_ERROR

```
#define AGS_JACK_DEVIN_ERROR (ags_jack_devin_error_quark())
```

enum AgsJackDevinError

Members

AGS_JACK_DEVIN_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_JACK_DEVIN

```
#define AGS_TYPE_JACK_DEVIN                 (ags_jack_devin_get_type())
```

struct AgsJackDevin

```
struct AgsJackDevin;
```

struct AgsJackDevinClass

```
struct AgsJackDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
"attack"                    gpointer
```

Attack of the buffer

Owner: AgsJackDevin

Flags: Read

Since: 3.0.0

The "bpm" property

```
"bpm"                       double
```

Beats per minute

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
"buffer"                    gpointer
```

The buffer

Owner: AgsJackDevin

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [1,44100]

Default value: 940

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The jack soundcard indentifier

Owner: AgsJackDevin

Flags: Read / Write

Default value: "ags-jack-devin-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "jack-client" property

```
"jack-client"                AgsJackClient~*
```

The assigned AgsJackClient

Owner: AgsJackDevin

Flags: Read / Write

Since: 3.0.0

The "jack-port" property

```
"jack-port"                  gpointer
```

The assigned AgsJackPort

Owner: AgsJackDevin

Flags: Read / Write

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"               guint
```

The pcm channel count

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                 guint
```

The samplerate

Owner: AgsJackDevin

Flags: Read / Write

Allowed values: [8000,96000]

Default value: 44100

Since: 3.0.0

8.14 AgsJackDevout

AgsJackDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_JACK_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_jack_devout_error_quark ()
gboolean	ags_jack_devout_test_flags ()
void	ags_jack_devout_set_flags ()
void	ags_jack_devout_unset_flags ()
void	ags_jack_devout_switch_buffer_flag ()
void	ags_jack_devout_adjust_delay_and_attack ()
void	ags_jack_devout_realloc_buffer ()
AgsJackDevout *	ags_jack_devout_new ()
#define	AGS_IS_JACK_DEVOUT()
#define	AGS_IS_JACK_DEVOUT_CLASS()
#define	AGS_JACK_DEVOUT()
#define	AGS_JACK_DEVOUT_CLASS()
#define	AGS_JACK_DEVOUT_GET_CLASS()
GType	ags_jack_devout_flags_get_type ()
GType	ags_jack_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
AgsJackClient *	jack-client	Read / Write
gpointer	jack-port	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_JACK_DEVOUT_DEFAULT_APP_BUFFER_SIZE
enum	AgsJackDevoutFlags
enum	AgsJackDevoutAppBufferMode
enum	AgsJackDevoutSyncFlags
#define	AGS_JACK_DEVOUT_ERROR
enum	AgsJackDevoutError
#define	AGS_TYPE_JACK_DEVOUT
struct	AgsJackDevout
struct	AgsJackDevoutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsJackDevoutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsJackDevout
```

Implemented Interfaces

AgsJackDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/jack/ags_jack_devout.h>
```

Description

AgsJackDevout represents a soundcard and supports output.

Functions

AGS_JACK_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_JACK_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsJackDevout *) obj)->obj_mutex))
```

ags_jack_devout_error_quark ()

```
GQuark
ags_jack_devout_error_quark ();
```

ags_jack_devout_test_flags ()

```
gboolean
ags_jack_devout_test_flags (AgsJackDevout *jack_devout,
                            guint flags);
```

Test *flags* to be set on *jack_devout* .

Parameters

jack_devout	the AgsJackDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_jack_devout_set_flags ()

```
void
ags_jack_devout_set_flags (AgsJackDevout *jack_devout,
                           guint flags);
```

Enable a feature of *jack_devout* .

Parameters

jack_devout	the AgsJackDevout	
flags	see AgsJackDevoutFlags	

Since: 3.0.0

ags_jack_devout_unset_flags ()

```
void
ags_jack_devout_unset_flags (AgsJackDevout *jack_devout,
                             guint flags);
```

Disable a feature of *jack_devout* .

Parameters

jack_devout	the AgsJackDevout	
flags	see AgsJackDevoutFlags	

Since: 3.0.0

ags_jack_devout_switch_buffer_flag ()

```
void
ags_jack_devout_switch_buffer_flag (AgsJackDevout *jack_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

jack_devout	an AgsJackDevout	

Since: 3.0.0

ags_jack_devout_adjust_delay_and_attack ()

```
void
ags_jack_devout_adjust_delay_and_attack
                                (AgsJackDevout *jack_devout);
```

Calculate delay and attack and reset it.

Parameters

jack_devout	the AgsJackDevout

Since: 3.0.0

ags_jack_devout_realloc_buffer ()

```
void
ags_jack_devout_realloc_buffer (AgsJackDevout *jack_devout);
```

Reallocate the internal audio buffer.

Parameters

jack_devout	the AgsJackDevout

Since: 3.0.0

ags_jack_devout_new ()

```
AgsJackDevout~*
ags_jack_devout_new ();
```

Creates a new instance of AgsJackDevout.

Returns

the new AgsJackDevout

Since: 3.0.0

AGS_IS_JACK_DEVOUT()

```
#define AGS_IS_JACK_DEVOUT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_JACK_DEVOUT))
```

AGS_IS_JACK_DEVOUT_CLASS()

```
#define AGS_IS_JACK_DEVOUT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_JACK_DEVOUT))
```

AGS_JACK_DEVOUT()

```
#define AGS_JACK_DEVOUT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_JACK_DEVOUT, AgsJackDevout))
```

AGS_JACK_DEVOUT_CLASS()

```
#define AGS_JACK_DEVOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_JACK_DEVOUT, AgsJackDevout))
```

AGS_JACK_DEVOUT_GET_CLASS()

```
#define AGS_JACK_DEVOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_JACK_DEVOUT, AgsJackDevoutClass))
```

ags_jack_devout_flags_get_type ()

```
GType
ags_jack_devout_flags_get_type ();
```

ags_jack_devout_get_type ()

```
GType
ags_jack_devout_get_type ();
```

Types and Values

AGS_JACK_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_JACK_DEVOUT_DEFAULT_APP_BUFFER_SIZE (4)
```

enum AgsJackDevoutFlags

Enum values to control the behavior or indicate internal state of AgsJackDevout by enable/disable as flags.

Members

AGS_JACK_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_JACK_DEVOUT_START_PLAY	playback start-ing
AGS_JACK_DEVOUT_PLAY	do play-back
AGS_JACK_DEVOUT_SHUTDOWN	stop play-back

AGS_JACK_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_JACK_DEVOUT_ATTACK_FIRST	use first attack, instead of second one

enum AgsJackDevoutAppBufferMode

Enum values to indicate internal state of AgsJackDevout application buffer by setting mode.

Members

AGS_JACK_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_JACK_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_JACK_DEVOUT_APP_BUFFER_2	ring-buffer 2
AGS_JACK_DEVOUT_APP_BUFFER_3	ring-buffer 3

enum AgsJackDevoutSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_JACK_DEVOUT_PASS_THROUGH	do not sync
AGS_JACK_DEVOUT_INITIAL_CALLBACK	initial call-back

AGS_JACK_DEVOUT_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_JACK_DEVOUT_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_JACK_DEVOUT_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_JACK_DEVOUT_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_JACK_DEVOUT_ERROR

```
#define AGS_JACK_DEVOUT_ERROR (ags_jack_devout_error_quark())
```

enum AgsJackDevoutError

Members

AGS_JACK_DEVOUT_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_JACK_DEVOUT

```
#define AGS_TYPE_JACK_DEVOUT                (ags_jack_devout_get_type())
```

struct AgsJackDevout

```
struct AgsJackDevout;
```

struct AgsJackDevoutClass

```
struct AgsJackDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsJackDevout

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsJackDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsJackDevout

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
  "buffer-size"               guint
```

The buffer size

Owner: AgsJackDevout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsJackDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The jack soundcard indentifier

Owner: AgsJackDevout

Flags: Read / Write

Default value: "ags-jack-devout-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsJackDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsJackDevout

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "jack-client" property

```
"jack-client"                AgsJackClient~*
```

The assigned AgsJackClient

Owner: AgsJackDevout

Flags: Read / Write

Since: 3.0.0

The "jack-port" property

```
"jack-port"                  gpointer
```

The assigned AgsJackPort

Owner: AgsJackDevout

Flags: Read / Write

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"               guint
```

The pcm channel count

Owner: AgsJackDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                 guint
```

The samplerate

Owner: AgsJackDevout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.0.0

8.15 AgsJackMidiin

AgsJackMidiin — Input from sequencer

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_JACK_MIDIIN_GET_OBJ_MUTEX()
GQuark	ags_jack_midiin_error_quark ()
gboolean	ags_jack_midiin_test_flags ()
void	ags_jack_midiin_set_flags ()
void	ags_jack_midiin_unset_flags ()
void	ags_jack_midiin_switch_buffer_flag ()
AgsJackMidiin *	ags_jack_midiin_new ()
#define	AGS_IS_JACK_MIDIIN()
#define	AGS_IS_JACK_MIDIIN_CLASS()
#define	AGS_JACK_MIDIIN()
#define	AGS_JACK_MIDIIN_CLASS()
#define	AGS_JACK_MIDIIN_GET_CLASS()
GType	ags_jack_midiin_flags_get_type ()
GType	ags_jack_midiin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
double	delay-factor	Read / Write
char *	device	Read / Write
AgsJackClient *	jack-client	Read / Write
AgsJackPort *	jack-port	Read / Write

Types and Values

#define	AGS_JACK_MIDIIN_DEFAULT_BUFFER_SIZE
enum	AgsJackMidiinFlags
enum	AgsJackMidiinAppBufferMode
enum	AgsJackMidiinSyncFlags
#define	AGS_JACK_MIDIIN_ERROR
enum	AgsJackMidiinError
#define	AGS_TYPE_JACK_MIDIIN
struct	AgsJackMidiin
struct	AgsJackMidiinClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsJackMidiinFlags
GObject
&#x2570;&#x2500;&#x2500; AgsJackMidiin
```

Implemented Interfaces

AgsJackMidiin implements AgsConnectable and AgsSequencer.

Includes

```
#include <ags/audio/jack/ags_jack_midiin.h>
```

Description

AgsJackMidiin represents a sequencer and supports midi input.

Functions

AGS_JACK_MIDIIN_GET_OBJ_MUTEX()

```
#define AGS_JACK_MIDIIN_GET_OBJ_MUTEX(obj) (&(((AgsJackMidiin *) obj)->obj_mutex))
```

ags_jack_midiin_error_quark ()

```
GQuark
ags_jack_midiin_error_quark ();
```

ags_jack_midiin_test_flags ()

```
gboolean
ags_jack_midiin_test_flags (AgsJackMidiin *jack_midiin,
                            guint flags);
```

Test *flags* to be set on *jack_midiin* .

Parameters

jack_midiin	the AgsJackMidiin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_jack_midiin_set_flags ()

```
void
ags_jack_midiin_set_flags (AgsJackMidiin *jack_midiin,
                           guint flags);
```

Enable a feature of *jack_midiin* .

Parameters

jack_midiin	the AgsJackMidiin	
flags	see AgsJackMidiinFlags	

Since: 3.0.0

ags_jack_midiin_unset_flags ()

```
void
ags_jack_midiin_unset_flags (AgsJackMidiin *jack_midiin,
                             guint flags);
```

Disable a feature of *jack_midiin*.

Parameters

jack_midiin	the AgsJackMidiin	
flags	see AgsJackMidiinFlags	

Since: 3.0.0

ags_jack_midiin_switch_buffer_flag ()

```
void
ags_jack_midiin_switch_buffer_flag (AgsJackMidiin *jack_midiin);
```

The buffer flag indicates the currently recorded buffer.

Parameters

jack_midiin	the AgsJackMidiin	

Since: 3.0.0

ags_jack_midiin_new ()

```
AgsJackMidiin~*
ags_jack_midiin_new ();
```

Creates a new instance of AgsJackMidiin.

Returns

the new AgsJackMidiin

Since: 3.0.0

AGS_IS_JACK_MIDIIN()

```
#define AGS_IS_JACK_MIDIIN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_JACK_MIDIIN))
```

AGS_IS_JACK_MIDIIN_CLASS()

```
#define AGS_IS_JACK_MIDIIN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_JACK_MIDIIN))
```

AGS_JACK_MIDIIN()

```
#define AGS_JACK_MIDIIN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_JACK_MIDIIN, AgsJackMidiin))
```

AGS_JACK_MIDIIN_CLASS()

```
#define AGS_JACK_MIDIIN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_JACK_MIDIIN, AgsJackMidiin))
```

AGS_JACK_MIDIIN_GET_CLASS()

```
#define AGS_JACK_MIDIIN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_JACK_MIDIIN, AgsJackMidiinClass))
```

ags_jack_midiin_flags_get_type ()

```
GType
ags_jack_midiin_flags_get_type ();
```

ags_jack_midiin_get_type ()

```
GType
ags_jack_midiin_get_type ();
```

Types and Values

AGS_JACK_MIDIIN_DEFAULT_BUFFER_SIZE

```
#define AGS_JACK_MIDIIN_DEFAULT_BUFFER_SIZE (256)
```

enum AgsJackMidiinFlags

Enum values to control the behavior or indicate internal state of AgsJackMidiin by enable/disable as flags.

Members

AGS_JACK_MIDIIN_INITIALIZED	recording is initialized
AGS_JACK_MIDIIN_START_RECORD	just started recording
AGS_JACK_MIDIIN_RECORD	is recording
AGS_JACK_MIDIIN_SHUTDOWN	stop recording
AGS_JACK_MIDIIN_NONBLOCKING	do non-blocking calls
AGS_JACK_MIDIIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsJackMidiinAppBufferMode

Enum values to indicate internal state of AgsJackMidiin application buffer by setting mode.

Members

AGS_JACK_MIDIIN_APP_BUFFER_0	ring-buffer 0
AGS_JACK_MIDIIN_APP_BUFFER_1	ring-buffer 1
AGS_JACK_MIDIIN_APP_BUFFER_2	ring-buffer 2
AGS_JACK_MIDIIN_APP_BUFFER_3	ring-buffer 3

enum AgsJackMidiinSyncFlags

Enum values to control the synchronization between sequencer and client.

Members

AGS_JACK_MIDIIN_PASS_THROUGH	do not sync
AGS_JACK_MIDIIN_INITIAL_CALLBACK	initial call-back
AGS_JACK_MIDIIN_CALLBACK_WAIT	sync wait, se-quencer con-di-tional lock
AGS_JACK_MIDIIN_CALLBACK_DONE	sync done, se-quencer con-di-tional lock
AGS_JACK_MIDIIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_JACK_MIDIIN_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_JACK_MIDIIN_ERROR

```
#define AGS_JACK_MIDIIN_ERROR (ags_jack_midiin_error_quark())
```

enum AgsJackMidiinError

Members

AGS_JACK_MIDIIN_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_JACK_MIDIIN

```
#define AGS_TYPE_JACK_MIDIIN                (ags_jack_midiin_get_type())
```

struct AgsJackMidiin

```
struct AgsJackMidiin;
```

struct AgsJackMidiinClass

```
struct AgsJackMidiinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsJackMidiin

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsJackMidiin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsJackMidiin

Flags: Read

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsJackMidiin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The JACK sequencer indentifier

Owner: AgsJackMidiin

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "jack-client" property

```
"jack-client"               AgsJackClient~*
```

The assigned AgsJackClient

Owner: AgsJackMidiin

Flags: Read / Write

Since: 3.0.0

The "jack-port" property

```
"jack-port"                 AgsJackPort~*
```

The assigned AgsJackPort

Owner: AgsJackMidiin

Flags: Read / Write

Since: 3.0.0

8.16 AgsPulseServer

AgsPulseServer — pulseaudio instance

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PULSE_SERVER_GET_OBJ_MUTEX()
gboolean	ags_pulse_server_test_flags ()
void	ags_pulse_server_set_flags ()
void	ags_pulse_server_unset_flags ()
GList *	ags_pulse_server_find_url ()
GObject *	ags_pulse_server_find_client ()
GObject *	ags_pulse_server_find_port ()
void	ags_pulse_server_add_client ()
void	ags_pulse_server_remove_client ()
void	ags_pulse_server_connect_client ()
void	ags_pulse_server_disconnect_client ()
void	ags_pulse_server_start_poll ()
AgsPulseServer *	ags_pulse_server_new ()
#define	AGS_IS_PULSE_SERVER()
#define	AGS_IS_PULSE_SERVER_CLASS()
#define	AGS_PULSE_SERVER()
#define	AGS_PULSE_SERVER_CLASS()
#define	AGS_PULSE_SERVER_GET_CLASS()
GType	ags_pulse_server_get_type ()

Properties

AgsPulseClient *	default-pulse-client	Read / Write
GObject *	default-soundcard	Read / Write
gpointer	pulse-client	Read / Write
char *	url	Read / Write

Types and Values

#define	AGS_TYPE_PULSE_SERVER
struct	AgsPulseServer
struct	AgsPulseServerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsPulseServer
```

Implemented Interfaces

AgsPulseServer implements AgsConnectable and AgsSoundServer.

Includes

```
#include <ags/audio/pulse/ags_pulse_server.h>
```

Description

The AgsPulseServer is an object to represent a running pulseaudio instance.

Functions

AGS_PULSE_SERVER_GET_OBJ_MUTEX()

```
#define AGS_PULSE_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsPulseServer *) obj)->obj_mutex))
```

ags_pulse_server_test_flags ()

```
gboolean
ags_pulse_server_test_flags (AgsPulseServer *pulse_server,
                             guint flags);
```

Test *flags* to be set on *pulse_server* .

Parameters

pulse_server	the AgsPulseServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_pulse_server_set_flags ()

```
void
ags_pulse_server_set_flags (AgsPulseServer *pulse_server,
                            guint flags);
```

Enable a feature of *pulse_server* .

Parameters

pulse_server	the AgsPulseServer	
flags	see AgsPulseServerFlags	

Since: 3.0.0

ags_pulse_server_unset_flags ()

```
void
ags_pulse_server_unset_flags (AgsPulseServer *pulse_server,
                              guint flags);
```

Disable a feature of *pulse_server* .

Parameters

pulse_server	the AgsPulseServer	
flags	see AgsPulseServerFlags	

Since: 3.0.0

ags_pulse_server_find_url ()

```
GList~*
ags_pulse_server_find_url (GList *pulse_server,
                           gchar *url);
```

Find AgsPulseServer by url.

Parameters

pulse_server	the GList containing AgsPulseServer.	*[element-type AgsAudio.PulseServer][transfer none]*
url	the url to find	

Returns

the GList containing a AgsPulseServer matching `url` or NULL.

[element-type AgsAudio.PulseServer][transfer none]

Since: 3.0.0

ags_pulse_server_find_client ()

```
GObject~*
ags_pulse_server_find_client (AgsPulseServer *pulse_server,
                              gchar *client_uuid);
```

Find AgsPulseClient by uuid.

Parameters

pulse_server	the AgsPulseServer	
client_uuid	the uuid to find	

Returns

the AgsPulseClient found or NULL.

[transfer none]

Since: 3.0.0

ags_pulse_server_find_port ()

```
GObject~*
ags_pulse_server_find_port (AgsPulseServer *pulse_server,
                            gchar *port_uuid);
```

Find AgsPulsePort by uuid.

Parameters

pulse_server	the AgsPulseServer	
port_uuid	the uuid to find	

Returns

the AgsPulsePort found or NULL.

[transfer none]

Since: 3.0.0

ags_pulse_server_add_client ()

```
void
ags_pulse_server_add_client (AgsPulseServer *pulse_server,
                             GObject *pulse_client);
```

Add *pulse_client* to *pulse_server*

Parameters

pulse_server	the AgsPulseServer	
pulse_client	the AgsPulseClient to add	

Since: 3.0.0

ags_pulse_server_remove_client ()

```
void
ags_pulse_server_remove_client (AgsPulseServer *pulse_server,
                                GObject *pulse_client);
```

Remove *pulse_client* to *pulse_server*

Parameters

pulse_server	the AgsPulseServer	
pulse_client	the AgsPulseClient to remove	

Since: 3.0.0

ags_pulse_server_connect_client ()

```
void
ags_pulse_server_connect_client (AgsPulseServer *pulse_server);
```

Connect all clients.

Parameters

pulse_server	the AgsPulseServer

Since: 3.0.0

ags_pulse_server_disconnect_client ()

```
void
ags_pulse_server_disconnect_client (AgsPulseServer *pulse_server);
```

Disconnect all clients.

Parameters

pulse_server	the AgsPulseServer

Since: 3.0.0

ags_pulse_server_start_poll ()

```
void
ags_pulse_server_start_poll (AgsPulseServer *pulse_server);
```

ags_pulse_server_new ()

```
AgsPulseServer~*
ags_pulse_server_new (gchar *url);
```

Create a new instance of AgsPulseServer.

Parameters

url	the URL as string

Returns

the new AgsPulseServer

Since: 3.0.0

AGS_IS_PULSE_SERVER()

```
#define AGS_IS_PULSE_SERVER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PULSE_SERVER))
```

AGS_IS_PULSE_SERVER_CLASS()

```
#define AGS_IS_PULSE_SERVER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PULSE_SERVER))
```

AGS_PULSE_SERVER()

```
#define AGS_PULSE_SERVER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PULSE_SERVER, AgsPulseServer))
```

AGS_PULSE_SERVER_CLASS()

```
#define AGS_PULSE_SERVER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_PULSE_SERVER, AgsPulseServer))
```

AGS_PULSE_SERVER_GET_CLASS()

```
#define AGS_PULSE_SERVER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_PULSE_SERVER, AgsPulseServerClass))
```

ags_pulse_server_get_type ()

```
GType
ags_pulse_server_get_type ();
```

Types and Values

AGS_TYPE_PULSE_SERVER

```
#define AGS_TYPE_PULSE_SERVER                 (ags_pulse_server_get_type())
```

struct AgsPulseServer

```
struct AgsPulseServer;
```

struct AgsPulseServerClass

```
struct AgsPulseServerClass {
  GObjectClass gobject;
};
```

Property Details

The "default-pulse-client" property

```
"default-pulse-client"      AgsPulseClient~*
```

The default pulse client.

Owner: AgsPulseServer

Flags: Read / Write

Since: 3.0.0

The "default-soundcard" property

```
"default-soundcard"         GObject~*
```

The default soundcard.

Owner: AgsPulseServer

Flags: Read / Write

Since: 3.0.0

The "pulse-client" property

```
"pulse-client"              gpointer
```

The pulse client list.

[transfer full]

Owner: AgsPulseServer

Flags: Read / Write

Since: 3.0.0

The "url" property

```
"url"                       char~*
```

The assigned URL.

Owner: AgsPulseServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

8.17 AgsPulseClient

AgsPulseClient — pulseaudio connection

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PULSE_CLIENT_GET_OBJ_MUTEX()
gboolean	ags_pulse_client_test_flags ()
void	ags_pulse_client_set_flags ()
void	ags_pulse_client_unset_flags ()
GList *	ags_pulse_client_find_uuid ()
GList *	ags_pulse_client_find ()
void	ags_pulse_client_open ()
void	ags_pulse_client_close ()
void	ags_pulse_client_add_device ()
void	ags_pulse_client_remove_device ()
void	ags_pulse_client_add_port ()
void	ags_pulse_client_remove_port ()
void	ags_pulse_client_activate ()
void	ags_pulse_client_deactivate ()
AgsPulseClient *	ags_pulse_client_new ()
#define	AGS_IS_PULSE_CLIENT()
#define	AGS_IS_PULSE_CLIENT_CLASS()
#define	AGS_PULSE_CLIENT()
#define	AGS_PULSE_CLIENT_CLASS()
#define	AGS_PULSE_CLIENT_GET_CLASS()
GType	ags_pulse_client_flags_get_type ()
GType	ags_pulse_client_get_type ()

Properties

char *	client-name	Read / Write
gpointer	device	Read / Write
gpointer	port	Read / Write
AgsPulseServer *	pulse-server	Read / Write

Types and Values

enum	AgsPulseClientFlags
#define	AGS_TYPE_PULSE_CLIENT
struct	AgsPulseClient
struct	AgsPulseClientClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsPulseClientFlags
GObject
&#x2570;&#x2500;&#x2500; AgsPulseClient
```

Implemented Interfaces

AgsPulseClient implements AgsConnectable.

Includes

```
#include <ags/audio/pulse/ags_pulse_client.h>
```

Description

The AgsPulseClient communicates with a pulseaudio instance.

Functions

AGS_PULSE_CLIENT_GET_OBJ_MUTEX()

```
#define AGS_PULSE_CLIENT_GET_OBJ_MUTEX(obj) (&(((AgsPulseClient *) obj)->obj_mutex))
```

ags_pulse_client_test_flags ()

```
gboolean
ags_pulse_client_test_flags (AgsPulseClient *pulse_client,
                             guint flags);
```

Test *flags* to be set on *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_pulse_client_set_flags ()

```
void
ags_pulse_client_set_flags (AgsPulseClient *pulse_client,
                            guint flags);
```

Enable a feature of *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
flags	see AgsPulseClientFlags	

Since: 3.0.0

ags_pulse_client_unset_flags ()

```
void
ags_pulse_client_unset_flags (AgsPulseClient *pulse_client,
                              guint flags);
```

Disable a feature of *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
flags	see AgsPulseClientFlags	

Since: 3.0.0

ags_pulse_client_find_uuid ()

```
GList~*
ags_pulse_client_find_uuid (GList *pulse_client,
                            gchar *client_uuid);
```

Finds next match of *client_uuid* in *pulse_client* .

Parameters

pulse_client	the GList containing AgsPulseClient.	*[element-type AgsAudio.PulseClient][transfer none]*
client_uuid	the client uuid to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.PulseClient][transfer none]

Since: 3.0.0

ags_pulse_client_find ()

```
GList~*
ags_pulse_client_find (GList *pulse_client,
                       gchar *client_name);
```

Finds next match of *client_name* in *pulse_client* .

Parameters

pulse_client	the GList containing AgsPulseClient.	*[element-type AgsAudio.PulseClient][transfer none]*
client_name	the client name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.PulseClient][transfer none]

Since: 3.0.0

ags_pulse_client_open ()

```
void
ags_pulse_client_open (AgsPulseClient *pulse_client,
                       gchar *client_name);
```

Open the pulseaudio client's connection and read uuid.

Parameters

pulse_client	the AgsPulseClient	
client_name	the client's name	

Since: 3.0.0

ags_pulse_client_close ()

```
void
ags_pulse_client_close (AgsPulseClient *pulse_client);
```

Close the pulseaudio client's connection.

Parameters

pulse_client	the AgsPulseClient	

Since: 3.0.0

ags_pulse_client_add_device ()

```
void
ags_pulse_client_add_device (AgsPulseClient *pulse_client,
                             GObject *pulse_device);
```

Add *pulse_device* to *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
pulse_device	an AgsPulseDevout or AgsPulseDevin	

Since: 3.0.0

ags_pulse_client_remove_device ()

```
void
ags_pulse_client_remove_device (AgsPulseClient *pulse_client,
                                GObject *pulse_device);
```

Remove *pulse_device* from *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
pulse_device	an AgsPulseDevout or AgsPulseDevin	

Since: 3.0.0

ags_pulse_client_add_port ()

```
void
ags_pulse_client_add_port (AgsPulseClient *pulse_client,
                           GObject *pulse_port);
```

Add *pulse_port* to *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
pulse_port	an AgsPulsePort	

Since: 3.0.0

ags_pulse_client_remove_port ()

```
void
ags_pulse_client_remove_port (AgsPulseClient *pulse_client,
                              GObject *pulse_port);
```

Remove *pulse_port* from *pulse_client* .

Parameters

pulse_client	the AgsPulseClient	
pulse_port	an AgsPulsePort	

Since: 3.0.0

ags_pulse_client_activate ()

```
void
ags_pulse_client_activate (AgsPulseClient *pulse_client);
```

Activate client.

Parameters

pulse_client	the AgsPulseClient

Since: 3.0.0

ags_pulse_client_deactivate ()

```
void
ags_pulse_client_deactivate (AgsPulseClient *pulse_client);
```

Deactivate client.

Parameters

pulse_client	the AgsPulseClient

Since: 3.0.0

ags_pulse_client_new ()

```
AgsPulseClient~*
ags_pulse_client_new (GObject *pulse_server);
```

Create a new instance of AgsPulseClient.

Parameters

pulse_server	the assigned AgsPulseServer

Returns

the new AgsPulseClient

Since: 3.0.0

AGS_IS_PULSE_CLIENT()

```
#define AGS_IS_PULSE_CLIENT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PULSE_CLIENT))
```

AGS_IS_PULSE_CLIENT_CLASS()

```
#define AGS_IS_PULSE_CLIENT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PULSE_CLIENT))
```

AGS_PULSE_CLIENT()

```
#define AGS_PULSE_CLIENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PULSE_CLIENT, AgsPulseClient))
```

AGS_PULSE_CLIENT_CLASS()

```
#define AGS_PULSE_CLIENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_PULSE_CLIENT, AgsPulseClient))
```

AGS_PULSE_CLIENT_GET_CLASS()

```
#define AGS_PULSE_CLIENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_PULSE_CLIENT, AgsPulseClientClass))
```

ags_pulse_client_flags_get_type ()

```
GType
ags_pulse_client_flags_get_type ();
```

ags_pulse_client_get_type ()

```
GType
ags_pulse_client_get_type ();
```

Types and Values

enum AgsPulseClientFlags

Enum values to control the behavior or indicate internal state of AgsPulseClient by enable/disable as flags.

Members

AGS_PULSE_CLIENT_ACTIVATED	the client was activated
AGS_PULSE_CLIENT_READY	the client is ready

AGS_TYPE_PULSE_CLIENT

```
#define AGS_TYPE_PULSE_CLIENT                (ags_pulse_client_get_type())
```

struct AgsPulseClient

```
struct AgsPulseClient;
```

struct AgsPulseClientClass

```
struct AgsPulseClientClass {
  GObjectClass gobject;
};
```

Property Details

The "client-name" property

```
  "client-name"              char~*
```

The pulseaudio client name.

Owner: AgsPulseClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "device" property

```
  "device"                   gpointer
```

The assigned devices.

[transfer full]

Owner: AgsPulseClient

Flags: Read / Write

Since: 3.0.0

The "port" property

```
  "port"                     gpointer
```

The assigned ports.

[transfer full]

Owner: AgsPulseClient

Flags: Read / Write

Since: 3.0.0

The "pulse-server" property

```
"pulse-server"              AgsPulseServer~*
```

The assigned AgsPulseServer.

Owner: AgsPulseClient

Flags: Read / Write

Since: 3.0.0

8.18 AgsPulsePort

AgsPulsePort — pulseaudio resource.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PULSE_PORT_GET_OBJ_MUTEX()
gboolean	ags_pulse_port_test_flags ()
void	ags_pulse_port_set_flags ()
void	ags_pulse_port_unset_flags ()
GList *	ags_pulse_port_find ()
void	ags_pulse_port_register ()
void	ags_pulse_port_unregister ()
guint	ags_pulse_port_get_fixed_size ()
void	ags_pulse_port_set_samplerate ()
void	ags_pulse_port_set_pcm_channels ()
void	ags_pulse_port_set_buffer_size ()
void	ags_pulse_port_set_format ()
void	ags_pulse_port_set_cache_buffer_size ()
guint	ags_pulse_port_get_latency ()
AgsPulsePort *	ags_pulse_port_new ()
#define	AGS_IS_PULSE_PORT()
#define	AGS_IS_PULSE_PORT_CLASS()
#define	AGS_PULSE_PORT()
#define	AGS_PULSE_PORT_CLASS()
#define	AGS_PULSE_PORT_GET_CLASS()
GType	ags_pulse_port_flags_get_type ()
GType	ags_pulse_port_get_type ()

Properties

char *	port-name	Read / Write
AgsPulseClient *	pulse-client	Read / Write
AgsPulseDevin *	pulse-devin	Read / Write
AgsPulseDevout *	pulse-devout	Read / Write

Types and Values

#define	AGS_PULSE_PORT_DEFAULT_CACHE_COUNT
#define	AGS_PULSE_PORT_DEFAULT_CACHE_BUFFER_SIZE
enum	AgsPulsePortFlags
#define	AGS_TYPE_PULSE_PORT
struct	AgsPulsePort
struct	AgsPulsePortClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsPulsePortFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsPulsePort
```

Implemented Interfaces

AgsPulsePort implements AgsConnectable.

Includes

```
#include <ags/audio/pulse/ags_pulse_port.h>
```

Description

The AgsPulsePort represents either a pulseaudio sequencer or soundcard to communicate with.

Functions

AGS_PULSE_PORT_GET_OBJ_MUTEX()

```
#define AGS_PULSE_PORT_GET_OBJ_MUTEX(obj) (&(((AgsPulsePort *) obj)->obj_mutex))
```

ags_pulse_port_test_flags ()

```
gboolean
ags_pulse_port_test_flags (AgsPulsePort *pulse_port,
                           guint flags);
```

Test *flags* to be set on *pulse_port* .

Parameters

pulse_port	the AgsPulsePort	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_pulse_port_set_flags ()

```
void
ags_pulse_port_set_flags (AgsPulsePort *pulse_port,
                          guint flags);
```

Enable a feature of *pulse_port* .

Parameters

pulse_port	the AgsPulsePort	
flags	see AgsPulsePortFlags	

Since: 3.0.0

ags_pulse_port_unset_flags ()

```
void
ags_pulse_port_unset_flags (AgsPulsePort *pulse_port,
                            guint flags);
```

Disable a feature of *pulse_port* .

Parameters

pulse_port	the AgsPulsePort	
flags	see AgsPulsePortFlags	

Since: 3.0.0

ags_pulse_port_find ()

```
GList~*
ags_pulse_port_find (GList *pulse_port,
                     gchar *port_name);
```

Finds next match of *port_name* in *pulse_port* .

Parameters

pulse_port	the GList containig AgsPulsePort.	*[element-type AgsAudio.PulsePort][transfer none]*
port_name	the port name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.PulsePort][transfer none]

Since: 3.0.0

ags_pulse_port_register ()

```
void
ags_pulse_port_register (AgsPulsePort *pulse_port,
                         gchar *port_name,
                         gboolean is_audio,
                         gboolean is_midi,
                         gboolean is_output);
```

Register a new pulseaudio port and read uuid. Creates a new AgsSequencer or AgsSoundcard object.

Parameters

pulse_port	the AgsPulsePort	
port_name	the name as string	
is_audio	if TRUE interpreted as audio port	
is_midi	if TRUE interpreted as midi port	
is_output	if TRUE port is acting as output, otherwise as input	

Since: 3.0.0

ags_pulse_port_unregister ()

```
void
ags_pulse_port_unregister (AgsPulsePort *pulse_port);
```

ags_pulse_port_get_fixed_size ()

```
guint
ags_pulse_port_get_fixed_size (AgsPulsePort *pulse_port);
```

ags_pulse_port_set_samplerate ()

```
void
ags_pulse_port_set_samplerate (AgsPulsePort *pulse_port,
                               guint samplerate);
```

ags_pulse_port_set_pcm_channels ()

```
void
ags_pulse_port_set_pcm_channels (AgsPulsePort *pulse_port,
                                 guint pcm_channels);
```

ags_pulse_port_set_buffer_size ()

```
void
ags_pulse_port_set_buffer_size (AgsPulsePort *pulse_port,
                                guint buffer_size);
```

ags_pulse_port_set_format ()

```
void
ags_pulse_port_set_format (AgsPulsePort *pulse_port,
                           guint format);
```

ags_pulse_port_set_cache_buffer_size ()

```
void
ags_pulse_port_set_cache_buffer_size (AgsPulsePort *pulse_port,
                                      guint cache_buffer_size);
```

ags_pulse_port_get_latency ()

```
guint
ags_pulse_port_get_latency (AgsPulsePort *pulse_port);
```

Gets latency.

Parameters

pulse_port	the AgsPulsePort

Since: 3.0.0

ags_pulse_port_new ()

```
AgsPulsePort~*
ags_pulse_port_new (GObject *pulse_client);
```

Create a new instance of AgsPulsePort.

Parameters

pulse_client	the AgsPulseClient assigned to

Returns

the new AgsPulsePort

Since: 3.0.0

AGS_IS_PULSE_PORT()

```
#define AGS_IS_PULSE_PORT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PULSE_PORT))
```

AGS_IS_PULSE_PORT_CLASS()

```
#define AGS_IS_PULSE_PORT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PULSE_PORT))
```

AGS_PULSE_PORT()

```
#define AGS_PULSE_PORT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PULSE_PORT, AgsPulsePort))
```

AGS_PULSE_PORT_CLASS()

```
#define AGS_PULSE_PORT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_PULSE_PORT, AgsPulsePort))
```

AGS_PULSE_PORT_GET_CLASS()

```
#define AGS_PULSE_PORT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_PULSE_PORT, AgsPulsePortClass))
```

ags_pulse_port_flags_get_type ()

```
GType
ags_pulse_port_flags_get_type ();
```

ags_pulse_port_get_type ()

```
GType
ags_pulse_port_get_type ();
```

Types and Values

AGS_PULSE_PORT_DEFAULT_CACHE_COUNT

```
#define AGS_PULSE_PORT_DEFAULT_CACHE_COUNT (4)
```

AGS_PULSE_PORT_DEFAULT_CACHE_BUFFER_SIZE

```
#define AGS_PULSE_PORT_DEFAULT_CACHE_BUFFER_SIZE (4096)
```

enum AgsPulsePortFlags

Enum values to control the behavior or indicate internal state of AgsPulsePort by enable/disable as flags.

Members

AGS_PULSE_PORT_REGISTERED	the port was registered
AGS_PULSE_PORT_IS_AUDIO	the port provides audio data
AGS_PULSE_PORT_IS_MIDI	the port provides midi data
AGS_PULSE_PORT_IS_OUTPUT	the port does output
AGS_PULSE_PORT_IS_INPUT	the port does input

AGS_TYPE_PULSE_PORT

```
#define AGS_TYPE_PULSE_PORT                (ags_pulse_port_get_type())
```

struct AgsPulsePort

```
struct AgsPulsePort;
```

struct AgsPulsePortClass

```
struct AgsPulsePortClass {
  GObjectClass gobject;
};
```

Property Details

The "port-name" property

```
"port-name"                 char~*
```

The pulse soundcard indentifier

Owner: AgsPulsePort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "pulse-client" property

```
"pulse-client"              AgsPulseClient~*
```

The assigned AgsPulseClient.

Owner: AgsPulsePort

Flags: Read / Write

Since: 3.0.0

The "pulse-devin" property

```
"pulse-devin"               AgsPulseDevin~*
```

The assigned AgsPulseDevout.

Owner: AgsPulsePort

Flags: Read / Write

Since: 3.0.0

The "pulse-devout" property

```
"pulse-devout"              AgsPulseDevout~*
```

The assigned AgsPulseDevout.

Owner: AgsPulsePort

Flags: Read / Write

Since: 3.0.0

8.19 AgsPulseDevin

AgsPulseDevin — Input from soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PULSE_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_pulse_devin_error_quark ()
gboolean	ags_pulse_devin_test_flags ()
void	ags_pulse_devin_set_flags ()
void	ags_pulse_devin_unset_flags ()
void	ags_pulse_devin_switch_buffer_flag ()
void	ags_pulse_devin_adjust_delay_and_attack ()
void	ags_pulse_devin_realloc_buffer ()
AgsPulseDevin *	ags_pulse_devin_new ()
#define	AGS_IS_PULSE_DEVIN()
#define	AGS_IS_PULSE_DEVIN_CLASS()
#define	AGS_PULSE_DEVIN()
#define	AGS_PULSE_DEVIN_CLASS()
#define	AGS_PULSE_DEVIN_GET_CLASS()
GType	ags_pulse_devin_flags_get_type ()
GType	ags_pulse_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
AgsPulseClient *	pulse-client	Read / Write
gpointer	pulse-port	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_PULSE_DEVIN_DEFAULT_APP_BUFFER_SIZE
enum	AgsPulseDevinFlags
enum	AgsPulseDevinAppBufferMode
enum	AgsPulseDevinSyncFlags
#define	AGS_PULSE_DEVIN_ERROR
enum	AgsPulseDevinError
#define	AGS_TYPE_PULSE_DEVIN
struct	AgsPulseDevin
struct	AgsPulseDevinClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsPulseDevinFlags
GObject
&#x2570;&#x2500;&#x2500; AgsPulseDevin
```

Implemented Interfaces

AgsPulseDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/pulse/ags_pulse_devin.h>
```

Description

AgsPulseDevin represents a soundcard and supports input.

Functions

AGS_PULSE_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_PULSE_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsPulseDevin *) obj)->obj_mutex))
```

ags_pulse_devin_error_quark ()

```
GQuark
ags_pulse_devin_error_quark ();
```

ags_pulse_devin_test_flags ()

```
gboolean
ags_pulse_devin_test_flags (AgsPulseDevin *pulse_devin,
                            guint flags);
```

Test *flags* to be set on *pulse_devin* .

Parameters

pulse_devin	the AgsPulseDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_pulse_devin_set_flags ()

```
void
ags_pulse_devin_set_flags (AgsPulseDevin *pulse_devin,
                           guint flags);
```

Enable a feature of *pulse_devin* .

Parameters

pulse_devin	the AgsPulseDevin	
flags	see AgsPulseDevinFlags	

Since: 3.0.0

ags_pulse_devin_unset_flags ()

```
void
ags_pulse_devin_unset_flags (AgsPulseDevin *pulse_devin,
                             guint flags);
```

Disable a feature of *pulse_devin* .

Parameters

pulse_devin	the AgsPulseDevin	
flags	see AgsPulseDevinFlags	

Since: 3.0.0

ags_pulse_devin_switch_buffer_flag ()

```
void
ags_pulse_devin_switch_buffer_flag (AgsPulseDevin *pulse_devin);
```

The buffer flag indicates the currently played buffer.

Parameters

pulse_devin	an AgsPulseDevin	

Since: 3.0.0

ags_pulse_devin_adjust_delay_and_attack ()

```
void
ags_pulse_devin_adjust_delay_and_attack
                                (AgsPulseDevin *pulse_devin);
```

Calculate delay and attack and reset it.

Parameters

pulse_devin	the AgsPulseDevin	

Since: 3.0.0

ags_pulse_devin_realloc_buffer ()

```
void
ags_pulse_devin_realloc_buffer (AgsPulseDevin *pulse_devin);
```

Reallocate the internal audio buffer.

Parameters

pulse_devin	the AgsPulseDevin

Since: 3.0.0

ags_pulse_devin_new ()

```
AgsPulseDevin~*
ags_pulse_devin_new ();
```

Creates a new instance of AgsPulseDevin.

Returns

the new AgsPulseDevin

Since: 3.0.0

AGS_IS_PULSE_DEVIN()

```
#define AGS_IS_PULSE_DEVIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PULSE_DEVIN))
```

AGS_IS_PULSE_DEVIN_CLASS()

```
#define AGS_IS_PULSE_DEVIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PULSE_DEVIN))
```

AGS_PULSE_DEVIN()

```
#define AGS_PULSE_DEVIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PULSE_DEVIN, AgsPulseDevin))
```

AGS_PULSE_DEVIN_CLASS()

```
#define AGS_PULSE_DEVIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_PULSE_DEVIN, AgsPulseDevin))
```

AGS_PULSE_DEVIN_GET_CLASS()

```
#define AGS_PULSE_DEVIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_PULSE_DEVIN, AgsPulseDevinClass))
```

ags_pulse_devin_flags_get_type ()

```
GType
ags_pulse_devin_flags_get_type ();
```

ags_pulse_devin_get_type ()

```
GType
ags_pulse_devin_get_type ();
```

Types and Values

AGS_PULSE_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_PULSE_DEVIN_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsPulseDevinFlags

Enum values to control the behavior or indicate internal state of AgsPulseDevin by enable/disable as flags.

Members

AGS_PULSE_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_PULSE_DEVIN_START_RECORD	capture start-ing
AGS_PULSE_DEVIN_RECORD	do cap-ture
AGS_PULSE_DEVIN_SHUTDOWN	stop cap-ture
AGS_PULSE_DEVIN_NONBLOCKING	do non-blocking calls

AGS_PULSE_DEVIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsPulseDevinAppBufferMode

Enum values to indicate internal state of AgsPulseDevin application buffer by setting mode.

Members

AGS_PULSE_DEVIN_APP_BUFFER_0	ring-buffer 0
AGS_PULSE_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_PULSE_DEVIN_APP_BUFFER_2	ring-buffer 2
AGS_PULSE_DEVIN_APP_BUFFER_3	ring-buffer 3
AGS_PULSE_DEVIN_APP_BUFFER_4	ring-buffer 4
AGS_PULSE_DEVIN_APP_BUFFER_5	ring-buffer 5
AGS_PULSE_DEVIN_APP_BUFFER_6	ring-buffer 6
AGS_PULSE_DEVIN_APP_BUFFER_7	ring-buffer 7

enum AgsPulseDevinSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_PULSE_DEVIN_PASS_THROUGH	do not sync

AGS_PULSE_DEVIN_INITIAL_CALLBACK	initial call-back
AGS_PULSE_DEVIN_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_PULSE_DEVIN_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_PULSE_DEVIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_PULSE_DEVIN_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_PULSE_DEVIN_ERROR

```
#define AGS_PULSE_DEVIN_ERROR (ags_pulse_devin_error_quark())
```

enum AgsPulseDevinError

Members

AGS_PULSE_DEVIN_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_PULSE_DEVIN

```
#define AGS_TYPE_PULSE_DEVIN                (ags_pulse_devin_get_type())
```

struct AgsPulseDevin

```
struct AgsPulseDevin;
```

struct AgsPulseDevinClass

```
struct AgsPulseDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsPulseDevin

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsPulseDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsPulseDevin

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
  "buffer-size"               guint
```

The buffer size

Owner: AgsPulseDevin

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsPulseDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The pulse soundcard indentifier

Owner: AgsPulseDevin

Flags: Read / Write

Default value: "ags-pulse-devin-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsPulseDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsPulseDevin

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsPulseDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "pulse-client" property

```
"pulse-client"              AgsPulseClient~*
```

The assigned AgsPulseClient

Owner: AgsPulseDevin

Flags: Read / Write

Since: 3.0.0

The "pulse-port" property

```
"pulse-port"                gpointer
```

The assigned AgsPulsePort

Owner: AgsPulseDevin

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsPulseDevin

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.0.0

8.20 AgsPulseDevout

AgsPulseDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PULSE_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_pulse_devout_error_quark ()
gboolean	ags_pulse_devout_test_flags ()
void	ags_pulse_devout_set_flags ()
void	ags_pulse_devout_unset_flags ()
void	ags_pulse_devout_switch_buffer_flag ()
void	ags_pulse_devout_adjust_delay_and_attack ()
void	ags_pulse_devout_realloc_buffer ()
AgsPulseDevout *	ags_pulse_devout_new ()
#define	AGS_IS_PULSE_DEVOUT()
#define	AGS_IS_PULSE_DEVOUT_CLASS()
#define	AGS_PULSE_DEVOUT()
#define	AGS_PULSE_DEVOUT_CLASS()
#define	AGS_PULSE_DEVOUT_GET_CLASS()
GType	ags_pulse_devout_flags_get_type ()
GType	ags_pulse_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
AgsPulseClient *	pulse-client	Read / Write
gpointer	pulse-port	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_PULSE_DEVOUT_DEFAULT_APP_BUFFER_SIZE
enum	AgsPulseDevoutFlags
enum	AgsPulseDevoutAppBufferMode
enum	AgsPulseDevoutSyncFlags
#define	AGS_PULSE_DEVOUT_ERROR
enum	AgsPulseDevoutError
#define	AGS_TYPE_PULSE_DEVOUT
struct	AgsPulseDevout
struct	AgsPulseDevoutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsPulseDevoutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsPulseDevout
```

Implemented Interfaces

AgsPulseDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/pulse/ags_pulse_devout.h>
```

Description

AgsPulseDevout represents a soundcard and supports output.

Functions

AGS_PULSE_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_PULSE_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsPulseDevout *) obj)->obj_mutex))
```

ags_pulse_devout_error_quark ()

```
GQuark
ags_pulse_devout_error_quark ();
```

ags_pulse_devout_test_flags ()

```
gboolean
ags_pulse_devout_test_flags (AgsPulseDevout *pulse_devout,
                             guint flags);
```

Test *flags* to be set on *pulse_devout* .

Parameters

pulse_devout	the AgsPulseDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_pulse_devout_set_flags ()

```
void
ags_pulse_devout_set_flags (AgsPulseDevout *pulse_devout,
                            guint flags);
```

Enable a feature of *pulse_devout* .

Parameters

pulse_devout	the AgsPulseDevout	
flags	see AgsPulseDevoutFlags	

Since: 3.0.0

ags_pulse_devout_unset_flags ()

```
void
ags_pulse_devout_unset_flags (AgsPulseDevout *pulse_devout,
                              guint flags);
```

Disable a feature of *pulse_devout* .

Parameters

pulse_devout	the AgsPulseDevout	
flags	see AgsPulseDevoutFlags	

Since: 3.0.0

ags_pulse_devout_switch_buffer_flag ()

```
void
ags_pulse_devout_switch_buffer_flag (AgsPulseDevout *pulse_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

pulse_devout	an AgsPulseDevout	

Since: 3.0.0

ags_pulse_devout_adjust_delay_and_attack ()

```
void
ags_pulse_devout_adjust_delay_and_attack
                                (AgsPulseDevout *pulse_devout);
```

Calculate delay and attack and reset it.

Parameters

pulse_devout	the AgsPulseDevout

Since: 3.0.0

ags_pulse_devout_realloc_buffer ()

```
void
ags_pulse_devout_realloc_buffer (AgsPulseDevout *pulse_devout);
```

Reallocate the internal audio buffer.

Parameters

pulse_devout	the AgsPulseDevout

Since: 3.0.0

ags_pulse_devout_new ()

```
AgsPulseDevout~*
ags_pulse_devout_new ();
```

Creates a new instance of AgsPulseDevout.

Returns

the new AgsPulseDevout

Since: 3.0.0

AGS_IS_PULSE_DEVOUT()

```
#define AGS_IS_PULSE_DEVOUT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PULSE_DEVOUT))
```

AGS_IS_PULSE_DEVOUT_CLASS()

```
#define AGS_IS_PULSE_DEVOUT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PULSE_DEVOUT))
```

AGS_PULSE_DEVOUT()

```
#define AGS_PULSE_DEVOUT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PULSE_DEVOUT, AgsPulseDevout))
```

AGS_PULSE_DEVOUT_CLASS()

```
#define AGS_PULSE_DEVOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_PULSE_DEVOUT, AgsPulseDevout))
```

AGS_PULSE_DEVOUT_GET_CLASS()

```
#define AGS_PULSE_DEVOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_PULSE_DEVOUT, AgsPulseDevoutClass))
```

ags_pulse_devout_flags_get_type ()

```
GType
ags_pulse_devout_flags_get_type ();
```

ags_pulse_devout_get_type ()

```
GType
ags_pulse_devout_get_type ();
```

Types and Values

AGS_PULSE_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_PULSE_DEVOUT_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsPulseDevoutFlags

Enum values to control the behavior or indicate internal state of AgsPulseDevout by enable/disable as flags.

Members

AGS_PULSE_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_PULSE_DEVOUT_START_PLAY	playback start-ing
AGS_PULSE_DEVOUT_PLAY	do play-back
AGS_PULSE_DEVOUT_SHUTDOWN	stop play-back

AGS_PULSE_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_PULSE_DEVOUT_ATTACK_FIRST	use first attack, instead of second one

enum AgsPulseDevoutAppBufferMode

Enum values to indicate internal state of AgsPulseDevout application buffer by setting mode.

Members

AGS_PULSE_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_PULSE_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_PULSE_DEVOUT_APP_BUFFER_2	ring-buffer 2
AGS_PULSE_DEVOUT_APP_BUFFER_3	ring-buffer 3
AGS_PULSE_DEVOUT_APP_BUFFER_4	ring-buffer 4
AGS_PULSE_DEVOUT_APP_BUFFER_5	ring-buffer 5
AGS_PULSE_DEVOUT_APP_BUFFER_6	ring-buffer 6
AGS_PULSE_DEVOUT_APP_BUFFER_7	ring-buffer 7

enum AgsPulseDevoutSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_PULSE_DEVOUT_PASS_THROUGH	do not sync
AGS_PULSE_DEVOUT_INITIAL_CALLBACK	initial call-back
AGS_PULSE_DEVOUT_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_PULSE_DEVOUT_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_PULSE_DEVOUT_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_PULSE_DEVOUT_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_PULSE_DEVOUT_ERROR

```
#define AGS_PULSE_DEVOUT_ERROR (ags_pulse_devout_error_quark())
```

enum AgsPulseDevoutError

Members

AGS_PULSE_DEVOUT_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_PULSE_DEVOUT

```
#define AGS_TYPE_PULSE_DEVOUT                (ags_pulse_devout_get_type())
```

struct AgsPulseDevout

```
struct AgsPulseDevout;
```

struct AgsPulseDevoutClass

```
struct AgsPulseDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsPulseDevout

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsPulseDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsPulseDevout

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size

Owner: AgsPulseDevout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"               double
```

tact

Owner: AgsPulseDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                     char~*
```

The pulse soundcard indentifier

Owner: AgsPulseDevout

Flags: Read / Write

Default value: "ags-pulse-devout-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"               guint
```

The dsp channel count

Owner: AgsPulseDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsPulseDevout

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsPulseDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "pulse-client" property

```
"pulse-client"              AgsPulseClient~*
```

The assigned AgsPulseClient

Owner: AgsPulseDevout

Flags: Read / Write

Since: 3.0.0

The "pulse-port" property

```
"pulse-port"                gpointer
```

The assigned AgsPulsePort

Owner: AgsPulseDevout

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsPulseDevout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.0.0

8.21 AgsGstreamerServer

AgsGstreamerServer — gstreamer instance

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_GSTREAMER_SERVER_GET_OBJ_MUTEX()
gboolean	ags_gstreamer_server_test_flags ()
void	ags_gstreamer_server_set_flags ()
void	ags_gstreamer_server_unset_flags ()
GList *	ags_gstreamer_server_find_url ()
GObject *	ags_gstreamer_server_find_client ()
GObject *	ags_gstreamer_server_find_port ()
void	ags_gstreamer_server_add_client ()
void	ags_gstreamer_server_remove_client ()
void	ags_gstreamer_server_connect_client ()
void	ags_gstreamer_server_disconnect_client ()
void	ags_gstreamer_server_start_poll ()
AgsGstreamerServer *	ags_gstreamer_server_new ()
#define	AGS_GSTREAMER_SERVER()
#define	AGS_GSTREAMER_SERVER_CLASS()
#define	AGS_GSTREAMER_SERVER_GET_CLASS()
#define	AGS_IS_GSTREAMER_SERVER()
#define	AGS_IS_GSTREAMER_SERVER_CLASS()
GType	ags_gstreamer_server_get_type ()

Properties

AgsGstreamerClient *	default-gstreamer-client	Read / Write
GObject *	default-soundcard	Read / Write
gpointer	gstreamer-client	Read / Write
char *	url	Read / Write

Types and Values

#define	AGS_TYPE_GSTREAMER_SERVER
struct	AgsGstreamerServer
struct	AgsGstreamerServerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsGstreamerServer
```

Implemented Interfaces

AgsGstreamerServer implements AgsConnectable and AgsSoundServer.

Includes

```
#include <ags/audio/gstreamer/ags_gstreamer_server.h>
```

Description

The AgsGstreamerServer is an object to represent a running gstreamer instance.

Functions

AGS_GSTREAMER_SERVER_GET_OBJ_MUTEX()

```
#define AGS_GSTREAMER_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsGstreamerServer *) obj)->obj_mutex) ↩
    )
```

ags_gstreamer_server_test_flags ()

```
gboolean
ags_gstreamer_server_test_flags (AgsGstreamerServer *gstreamer_server,
                                 guint flags);
```

Test *flags* to be set on *gstreamer_server* .

Parameters

gstreamer_server	the AgsGstreamerServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.6.0

ags_gstreamer_server_set_flags ()

```
void
ags_gstreamer_server_set_flags (AgsGstreamerServer *gstreamer_server,
                                guint flags);
```

Enable a feature of *gstreamer_server* .

Parameters

gstreamer_server	the AgsGstreamerServer	
flags	see AgsGstreamerServerFlags	

Since: 3.6.0

ags_gstreamer_server_unset_flags ()

```
void
ags_gstreamer_server_unset_flags (AgsGstreamerServer *gstreamer_server,
                                  guint flags);
```

Disable a feature of *gstreamer_server* .

Parameters

gstreamer_server	the AgsGstreamerServer	
flags	see AgsGstreamerServerFlags	

Since: 3.6.0

ags_gstreamer_server_find_url ()

```
GList~*
ags_gstreamer_server_find_url (GList *gstreamer_server,
                               gchar *url);
```

Find AgsGstreamerServer by url.

Parameters

gstreamer_server	the GList containing AgsGstreamerServer.	*[element-type AgsAudio.GstreamerServer][transfer none]*
url	the url to find	

Returns

the GList containing a AgsGstreamerServer matching *url* or NULL.

[element-type AgsAudio.GstreamerServer][transfer none]

Since: 3.6.0

ags_gstreamer_server_find_client ()

```
GObject~*
ags_gstreamer_server_find_client (AgsGstreamerServer *gstreamer_server,
                                  gchar *client_uuid);
```

Find AgsGstreamerClient by uuid.

Parameters

gstreamer_server	the AgsGstreamerServer	
client_uuid	the uuid to find	

Returns

the AgsGstreamerClient found or NULL.

[transfer none]

Since: 3.6.0

ags_gstreamer_server_find_port ()

```
GObject~*
ags_gstreamer_server_find_port (AgsGstreamerServer *gstreamer_server,
                                gchar *port_uuid);
```

Find AgsGstreamerPort by uuid.

Parameters

gstreamer_server	the AgsGstreamerServer	
port_uuid	the uuid to find	

Returns

the AgsGstreamerPort found or NULL.

[transfer none]

Since: 3.6.0

ags_gstreamer_server_add_client ()

```
void
ags_gstreamer_server_add_client (AgsGstreamerServer *gstreamer_server,
                                 GObject *gstreamer_client);
```

Add *gstreamer_client* to *gstreamer_server*

Parameters

gstreamer_server	the AgsGstreamerServer	
gstreamer_client	the AgsGstreamerClient to add	

Since: 3.6.0

ags_gstreamer_server_remove_client ()

```
void
ags_gstreamer_server_remove_client (AgsGstreamerServer *gstreamer_server,
                                    GObject *gstreamer_client);
```

Remove *gstreamer_client* to *gstreamer_server*

Parameters

gstreamer_server	the AgsGstreamerServer	
gstreamer_client	the AgsGstreamerClient to remove	

Since: 3.6.0

ags_gstreamer_server_connect_client ()

```
void
ags_gstreamer_server_connect_client (AgsGstreamerServer *gstreamer_server);
```

Connect all clients.

Parameters

gstreamer_server	the AgsGstreamerServer	

Since: 3.6.0

ags_gstreamer_server_disconnect_client ()

```
void
ags_gstreamer_server_disconnect_client
                                (AgsGstreamerServer *gstreamer_server);
```

Disconnect all clients.

Parameters

gstreamer_server	the AgsGstreamerServer	

Since: 3.6.0

ags_gstreamer_server_start_poll ()

```
void
ags_gstreamer_server_start_poll (AgsGstreamerServer *gstreamer_server);
```

ags_gstreamer_server_new ()

```
AgsGstreamerServer~*
ags_gstreamer_server_new (gchar *url);
```

Create a new instance of AgsGstreamerServer.

Parameters

url	the URL as string

Returns

the new AgsGstreamerServer

Since: 3.6.0

AGS_GSTREAMER_SERVER()

```
#define AGS_GSTREAMER_SERVER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GSTREAMER_SERVER, AgsGstreamerServer))
```

AGS_GSTREAMER_SERVER_CLASS()

```
#define AGS_GSTREAMER_SERVER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_GSTREAMER_SERVER, AgsGstreamerServer))
```

AGS_GSTREAMER_SERVER_GET_CLASS()

```
#define AGS_GSTREAMER_SERVER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_GSTREAMER_SERVER, AgsGstreamerServerClass))
```

AGS_IS_GSTREAMER_SERVER()

```
#define AGS_IS_GSTREAMER_SERVER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_GSTREAMER_SERVER))
```

AGS_IS_GSTREAMER_SERVER_CLASS()

```
#define AGS_IS_GSTREAMER_SERVER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_GSTREAMER_SERVER))
```

ags_gstreamer_server_get_type ()

```
GType
ags_gstreamer_server_get_type ();
```

Types and Values

AGS_TYPE_GSTREAMER_SERVER

```
#define AGS_TYPE_GSTREAMER_SERVER                (ags_gstreamer_server_get_type())
```

struct AgsGstreamerServer

```
struct AgsGstreamerServer;
```

struct AgsGstreamerServerClass

```
struct AgsGstreamerServerClass {
  GObjectClass gobject;
};
```

Property Details

The "default-gstreamer-client" property

```
  "default-gstreamer-client" AgsGstreamerClient~*
```

The default gstreamer client.

Owner: AgsGstreamerServer

Flags: Read / Write

Since: 3.6.0

The "default-soundcard" property

```
  "default-soundcard"        GObject~*
```

The default soundcard.

Owner: AgsGstreamerServer

Flags: Read / Write

Since: 3.6.0

The "gstreamer-client" property

```
  "gstreamer-client"         gpointer
```

The gstreamer client list.

[transfer full]

Owner: AgsGstreamerServer

Flags: Read / Write

Since: 3.6.0

The "url" property

```
"url"                    char~*
```

The assigned URL.

Owner: AgsGstreamerServer

Flags: Read / Write

Default value: NULL

Since: 3.6.0

8.22 AgsGstreamerClient

AgsGstreamerClient — gstreamer connection

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_GSTREAMER_CLIENT_GET_OBJ_MUTEX()
gboolean	ags_gstreamer_client_test_flags ()
void	ags_gstreamer_client_set_flags ()
void	ags_gstreamer_client_unset_flags ()
GList *	ags_gstreamer_client_find_uuid ()
GList *	ags_gstreamer_client_find ()
void	ags_gstreamer_client_open ()
void	ags_gstreamer_client_close ()
void	ags_gstreamer_client_add_device ()
void	ags_gstreamer_client_remove_device ()
void	ags_gstreamer_client_add_port ()
void	ags_gstreamer_client_remove_port ()
void	ags_gstreamer_client_activate ()
void	ags_gstreamer_client_deactivate ()
AgsGstreamerClient *	ags_gstreamer_client_new ()
#define	AGS_GSTREAMER_CLIENT()
#define	AGS_GSTREAMER_CLIENT_CLASS()
#define	AGS_GSTREAMER_CLIENT_GET_CLASS()
#define	AGS_IS_GSTREAMER_CLIENT()
#define	AGS_IS_GSTREAMER_CLIENT_CLASS()
GType	ags_gstreamer_client_flags_get_type ()
GType	ags_gstreamer_client_get_type ()

Properties

char *	client-name	Read / Write
gpointer	device	Read / Write
AgsGstreamerServer *	gstreamer-server	Read / Write
gpointer	port	Read / Write

Types and Values

enum	AgsGstreamerClientFlags
#define	AGS_TYPE_GSTREAMER_CLIENT
struct	AgsGstreamerClient
struct	AgsGstreamerClientClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsGstreamerClientFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsGstreamerClient
```

Implemented Interfaces

AgsGstreamerClient implements AgsConnectable.

Includes

```
#include <ags/audio/gstreamer/ags_gstreamer_client.h>
```

Description

The AgsGstreamerClient communicates with a gstreamer instance.

Functions

AGS_GSTREAMER_CLIENT_GET_OBJ_MUTEX()

```
#define AGS_GSTREAMER_CLIENT_GET_OBJ_MUTEX(obj) (&(((AgsGstreamerClient *) obj)->obj_mutex) ↩
    )
```

ags_gstreamer_client_test_flags ()

```
gboolean
ags_gstreamer_client_test_flags (AgsGstreamerClient *gstreamer_client,
                                 guint flags);
```

Test *flags* to be set on *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.6.0

ags_gstreamer_client_set_flags ()

```
void
ags_gstreamer_client_set_flags (AgsGstreamerClient *gstreamer_client,
                                guint flags);
```

Enable a feature of *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
flags	see AgsGstreamerClientFlags	

Since: 3.6.0

ags_gstreamer_client_unset_flags ()

```
void
ags_gstreamer_client_unset_flags (AgsGstreamerClient *gstreamer_client,
                                  guint flags);
```

Disable a feature of *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
flags	see AgsGstreamerClientFlags	

Since: 3.6.0

ags_gstreamer_client_find_uuid ()

```
GList~*
ags_gstreamer_client_find_uuid (GList *gstreamer_client,
                                gchar *client_uuid);
```

Finds next match of *client_uuid* in *gstreamer_client* .

Parameters

gstreamer_client	the GList containing AgsGstreamerClient.	*[element-type AgsAudio.GstreamerClient][transfer none]*
client_uuid	the client uuid to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.GstreamerClient][transfer none]

Since: 3.6.0

ags_gstreamer_client_find ()

```
GList~*
ags_gstreamer_client_find (GList *gstreamer_client,
                           gchar *client_name);
```

Finds next match of *client_name* in *gstreamer_client* .

Parameters

gstreamer_client	the GList containing AgsGstreamerClient.	*[element-type AgsAudio.GstreamerClient][transfer none]*
client_name	the client name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.GstreamerClient][transfer none]

Since: 3.6.0

ags_gstreamer_client_open ()

```
void
ags_gstreamer_client_open (AgsGstreamerClient *gstreamer_client,
                           gchar *client_name);
```

Open the gstreamer client's connection and read uuid.

Parameters

gstreamer_client	the AgsGstreamerClient	
client_name	the client's name	

Since: 3.6.0

ags_gstreamer_client_close ()

```
void
ags_gstreamer_client_close (AgsGstreamerClient *gstreamer_client);
```

Close the gstreamer client's connection.

Parameters

gstreamer_client	the AgsGstreamerClient	

Since: 3.6.0

ags_gstreamer_client_add_device ()

```
void
ags_gstreamer_client_add_device (AgsGstreamerClient *gstreamer_client,
                                 GObject *gstreamer_device);
```

Add *gstreamer_device* to *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
gstreamer_device	an AgsGstreamerDevout or AgsGstreamerDevin	

Since: 3.6.0

ags_gstreamer_client_remove_device ()

```
void
ags_gstreamer_client_remove_device (AgsGstreamerClient *gstreamer_client,
                                    GObject *gstreamer_device);
```

Remove *gstreamer_device* from *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
gstreamer_device	an AgsGstreamerDevout or AgsGstreamerDevin	

Since: 3.6.0

ags_gstreamer_client_add_port ()

```
void
ags_gstreamer_client_add_port (AgsGstreamerClient *gstreamer_client,
                               GObject *gstreamer_port);
```

Add *gstreamer_port* to *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
gstreamer_port	an AgsGstreamerPort	

Since: 3.6.0

ags_gstreamer_client_remove_port ()

```
void
ags_gstreamer_client_remove_port (AgsGstreamerClient *gstreamer_client,
                                  GObject *gstreamer_port);
```

Remove *gstreamer_port* from *gstreamer_client* .

Parameters

gstreamer_client	the AgsGstreamerClient	
gstreamer_port	an AgsGstreamerPort	

Since: 3.6.0

ags_gstreamer_client_activate ()

```
void
ags_gstreamer_client_activate (AgsGstreamerClient *gstreamer_client);
```

Activate client.

Parameters

gstreamer_client	the AgsGstreamerClient	

Since: 3.6.0

ags_gstreamer_client_deactivate ()

```
void
ags_gstreamer_client_deactivate (AgsGstreamerClient *gstreamer_client);
```

Deactivate client.

Parameters

gstreamer_client	the AgsGstreamerClient	

Since: 3.6.0

ags_gstreamer_client_new ()

```
AgsGstreamerClient~*
ags_gstreamer_client_new (GObject *gstreamer_server);
```

Create a new instance of AgsGstreamerClient.

Parameters

gstreamer_server	the assigned AgsGstreamerServer	

Returns

the new AgsGstreamerClient

Since: 3.6.0

AGS_GSTREAMER_CLIENT()

```
#define AGS_GSTREAMER_CLIENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GSTREAMER_CLIENT, AgsGstreamerClient))
```

AGS_GSTREAMER_CLIENT_CLASS()

```
#define AGS_GSTREAMER_CLIENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_GSTREAMER_CLIENT, AgsGstreamerClient))
```

AGS_GSTREAMER_CLIENT_GET_CLASS()

```
#define AGS_GSTREAMER_CLIENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_GSTREAMER_CLIENT, AgsGstreamerClientClass))
```

AGS_IS_GSTREAMER_CLIENT()

```
#define AGS_IS_GSTREAMER_CLIENT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_GSTREAMER_CLIENT))
```

AGS_IS_GSTREAMER_CLIENT_CLASS()

```
#define AGS_IS_GSTREAMER_CLIENT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_GSTREAMER_CLIENT))
```

ags_gstreamer_client_flags_get_type ()

```
GType
ags_gstreamer_client_flags_get_type ();
```

ags_gstreamer_client_get_type ()

```
GType
ags_gstreamer_client_get_type ();
```

Types and Values

enum AgsGstreamerClientFlags

Enum values to control the behavior or indicate internal state of AgsGstreamerClient by enable/disable as flags.

Members

AGS_GSTREAMER_CLIENT_ACTIVATED	the client was activated
AGS_GSTREAMER_CLIENT_READY	the client is ready

AGS_TYPE_GSTREAMER_CLIENT

```
#define AGS_TYPE_GSTREAMER_CLIENT                (ags_gstreamer_client_get_type())
```

struct AgsGstreamerClient

```
struct AgsGstreamerClient;
```

struct AgsGstreamerClientClass

```
struct AgsGstreamerClientClass {
  GObjectClass gobject;
};
```

Property Details

The "client-name" property

```
  "client-name"              char~*
```

The gstreamer client name.

Owner: AgsGstreamerClient

Flags: Read / Write

Default value: NULL

Since: 3.6.0

The "device" property

```
  "device"                   gpointer
```

The assigned devices.

[transfer full]

Owner: AgsGstreamerClient

Flags: Read / Write

Since: 3.6.0

The "gstreamer-server" property

```
"gstreamer-server"          AgsGstreamerServer~*
```

The assigned AgsGstreamerServer.

Owner: AgsGstreamerClient

Flags: Read / Write

Since: 3.6.0

The "port" property

```
"port"                      gpointer
```

The assigned ports.

[transfer full]

Owner: AgsGstreamerClient

Flags: Read / Write

Since: 3.6.0

8.23 AgsGstreamerPort

AgsGstreamerPort — gstreamer resource.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_GSTREAMER_PORT_GET_OBJ_MUTEX()
gboolean	ags_gstreamer_port_test_flags ()
void	ags_gstreamer_port_set_flags ()
void	ags_gstreamer_port_unset_flags ()
GList *	ags_gstreamer_port_find ()
void	ags_gstreamer_port_register ()
void	ags_gstreamer_port_unregister ()
guint	ags_gstreamer_port_get_fixed_size ()
void	ags_gstreamer_port_set_samplerate ()
void	ags_gstreamer_port_set_pcm_channels ()
void	ags_gstreamer_port_set_buffer_size ()
void	ags_gstreamer_port_set_format ()
void	ags_gstreamer_port_set_cache_buffer_size ()
guint	ags_gstreamer_port_get_latency ()
AgsGstreamerPort *	ags_gstreamer_port_new ()
#define	AGS_GSTREAMER_PORT()
#define	AGS_GSTREAMER_PORT_CLASS()
#define	AGS_GSTREAMER_PORT_GET_CLASS()
#define	AGS_IS_GSTREAMER_PORT()

#define	AGS_IS_GSTREAMER_PORT_CLASS()
GType	ags_gstreamer_port_flags_get_type ()
GType	ags_gstreamer_port_get_type ()

Properties

AgsGstreamerClient *	gstreamer-client	Read / Write
AgsGstreamerDevin *	gstreamer-devin	Read / Write
AgsGstreamerDevout *	gstreamer-devout	Read / Write
char *	port-name	Read / Write

Types and Values

#define	AGS_GSTREAMER_PORT_DEFAULT_CACHE_BUFFER_SIZE
enum	AgsGstreamerPortFlags
#define	AGS_TYPE_GSTREAMER_PORT
struct	AgsGstreamerPort
struct	AgsGstreamerPortClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsGstreamerPortFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsGstreamerPort
```

Implemented Interfaces

AgsGstreamerPort implements AgsConnectable.

Includes

```
#include <ags/audio/gstreamer/ags_gstreamer_port.h>
```

Description

The AgsGstreamerPort represents either a gstreamer sequencer or soundcard to communicate with.

Functions

AGS_GSTREAMER_PORT_GET_OBJ_MUTEX()

```
#define AGS_GSTREAMER_PORT_GET_OBJ_MUTEX(obj) (&(((AgsGstreamerPort *) obj)->obj_mutex))
```

ags_gstreamer_port_test_flags ()

```
gboolean
ags_gstreamer_port_test_flags (AgsGstreamerPort *gstreamer_port,
                               guint flags);
```

Test *flags* to be set on *gstreamer_port* .

Parameters

gstreamer_port	the AgsGstreamerPort	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.6.0

ags_gstreamer_port_set_flags ()

```
void
ags_gstreamer_port_set_flags (AgsGstreamerPort *gstreamer_port,
                              guint flags);
```

Enable a feature of *gstreamer_port* .

Parameters

gstreamer_port	the AgsGstreamerPort	
flags	see AgsGstreamerPortFlags	

Since: 3.6.0

ags_gstreamer_port_unset_flags ()

```
void
ags_gstreamer_port_unset_flags (AgsGstreamerPort *gstreamer_port,
                                guint flags);
```

Disable a feature of *gstreamer_port* .

Parameters

gstreamer_port	the AgsGstreamerPort	
flags	see AgsGstreamerPortFlags	

Since: 3.6.0

ags_gstreamer_port_find ()

```
GList~*
ags_gstreamer_port_find (GList *gstreamer_port,
                         gchar *port_name);
```

Finds next match of *port_name* in *gstreamer_port* .

Parameters

gstreamer_port	the GList containig AgsGstreamerPort.	*[element-type AgsAudio.GstreamerPort][transfer none]*
port_name	the port name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.GstreamerPort][transfer none]

Since: 3.6.0

ags_gstreamer_port_register ()

```
void
ags_gstreamer_port_register (AgsGstreamerPort *gstreamer_port,
                             gchar *port_name,
                             gboolean is_audio,
                             gboolean is_midi,
                             gboolean is_output);
```

Register a new gstreamer port and read uuid. Creates a new AgsSequencer or AgsSoundcard object.

Parameters

gstreamer_port	the AgsGstreamerPort	
port_name	the name as string	
is_audio	if TRUE interpreted as audio port	
is_midi	if TRUE interpreted as midi port	
is_output	if TRUE port is acting as output, otherwise as input	

Since: 3.6.0

ags_gstreamer_port_unregister ()

```
void
ags_gstreamer_port_unregister (AgsGstreamerPort *gstreamer_port);
```

ags_gstreamer_port_get_fixed_size ()

```
guint
ags_gstreamer_port_get_fixed_size (AgsGstreamerPort *gstreamer_port);
```

ags_gstreamer_port_set_samplerate ()

```
void
ags_gstreamer_port_set_samplerate (AgsGstreamerPort *gstreamer_port,
                                   guint samplerate);
```

ags_gstreamer_port_set_pcm_channels ()

```
void
ags_gstreamer_port_set_pcm_channels (AgsGstreamerPort *gstreamer_port,
                                     guint pcm_channels);
```

ags_gstreamer_port_set_buffer_size ()

```
void
ags_gstreamer_port_set_buffer_size (AgsGstreamerPort *gstreamer_port,
                                    guint buffer_size);
```

ags_gstreamer_port_set_format ()

```
void
ags_gstreamer_port_set_format (AgsGstreamerPort *gstreamer_port,
                               guint format);
```

ags_gstreamer_port_set_cache_buffer_size ()

```
void
ags_gstreamer_port_set_cache_buffer_size
                               (AgsGstreamerPort *gstreamer_port,
                                guint cache_buffer_size);
```

ags_gstreamer_port_get_latency ()

```
guint
ags_gstreamer_port_get_latency (AgsGstreamerPort *gstreamer_port);
```

Gets latency.

Parameters

gstreamer_port	the AgsGstreamerPort

Since: 3.6.0

ags_gstreamer_port_new ()

```
AgsGstreamerPort~*
ags_gstreamer_port_new (GObject *gstreamer_client);
```

Create a new instance of AgsGstreamerPort.

Parameters

gstreamer_client	the AgsGstreamerClient assigned to

Returns

the new AgsGstreamerPort

Since: 3.6.0

AGS_GSTREAMER_PORT()

```
#define AGS_GSTREAMER_PORT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GSTREAMER_PORT, AgsGstreamerPort))
```

AGS_GSTREAMER_PORT_CLASS()

```
#define AGS_GSTREAMER_PORT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_GSTREAMER_PORT, AgsGstreamerPort))
```

AGS_GSTREAMER_PORT_GET_CLASS()

```
#define AGS_GSTREAMER_PORT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_GSTREAMER_PORT, AgsGstreamerPortClass))
```

AGS_IS_GSTREAMER_PORT()

```
#define AGS_IS_GSTREAMER_PORT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_GSTREAMER_PORT))
```

AGS_IS_GSTREAMER_PORT_CLASS()

```
#define AGS_IS_GSTREAMER_PORT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_GSTREAMER_PORT))
```

ags_gstreamer_port_flags_get_type ()

```
GType
ags_gstreamer_port_flags_get_type ();
```

ags_gstreamer_port_get_type ()

```
GType
ags_gstreamer_port_get_type ();
```

Types and Values

AGS_GSTREAMER_PORT_DEFAULT_CACHE_BUFFER_SIZE

```
#define AGS_GSTREAMER_PORT_DEFAULT_CACHE_BUFFER_SIZE (4096)
```

enum AgsGstreamerPortFlags

Enum values to control the behavior or indicate internal state of AgsGstreamerPort by enable/disable as flags.

Members

AGS_GSTREAMER_PORT_REGISTERED	the port was registered
AGS_GSTREAMER_PORT_IS_AUDIO	the port provides audio data
AGS_GSTREAMER_PORT_IS_MIDI	the port provides midi data
AGS_GSTREAMER_PORT_IS_OUTPUT	the port does output
AGS_GSTREAMER_PORT_IS_INPUT	the port does input

AGS_TYPE_GSTREAMER_PORT

```
#define AGS_TYPE_GSTREAMER_PORT                (ags_gstreamer_port_get_type())
```

struct AgsGstreamerPort

```
struct AgsGstreamerPort;
```

struct AgsGstreamerPortClass

```
struct AgsGstreamerPortClass {
  GObjectClass gobject;
};
```

Property Details

The "gstreamer-client" property

```
  "gstreamer-client"          AgsGstreamerClient~*
```

The assigned AgsGstreamerClient.

Owner: AgsGstreamerPort

Flags: Read / Write

Since: 3.6.0

The "gstreamer-devin" property

```
  "gstreamer-devin"           AgsGstreamerDevin~*
```

The assigned AgsGstreamerDevout.

Owner: AgsGstreamerPort

Flags: Read / Write

Since: 3.6.0

The "gstreamer-devout" property

```
  "gstreamer-devout"          AgsGstreamerDevout~*
```

The assigned AgsGstreamerDevout.

Owner: AgsGstreamerPort

Flags: Read / Write

Since: 3.6.0

The "port-name" property

```
  "port-name"                 char~*
```

The gstreamer soundcard indentifier

Owner: AgsGstreamerPort

Flags: Read / Write

Default value: NULL

Since: 3.6.0

8.24 AgsGstreamerDevin

AgsGstreamerDevin — Input from soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_GSTREAMER_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_gstreamer_devin_error_quark ()
gboolean	ags_gstreamer_devin_test_flags ()
void	ags_gstreamer_devin_set_flags ()
void	ags_gstreamer_devin_unset_flags ()
void	ags_gstreamer_devin_switch_buffer_flag ()
void	ags_gstreamer_devin_adjust_delay_and_attack ()
void	ags_gstreamer_devin_realloc_buffer ()
AgsGstreamerDevin *	ags_gstreamer_devin_new ()
#define	AGS_GSTREAMER_DEVIN()
#define	AGS_GSTREAMER_DEVIN_CLASS()
#define	AGS_GSTREAMER_DEVIN_GET_CLASS()
#define	AGS_IS_GSTREAMER_DEVIN()
#define	AGS_IS_GSTREAMER_DEVIN_CLASS()
GType	ags_gstreamer_devin_flags_get_type ()
GType	ags_gstreamer_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
AgsGstreamerClient *	gstreamer-client	Read / Write
gpointer	gstreamer-port	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_GSTREAMER_DEVIN_DEFAULT_APP_BUFFER_SIZE
enum	AgsGstreamerDevinFlags
enum	AgsGstreamerDevinAppBufferMode
enum	AgsGstreamerDevinSyncFlags
#define	AGS_GSTREAMER_DEVIN_ERROR
enum	AgsGstreamerDevinError
#define	AGS_TYPE_GSTREAMER_DEVIN

struct	AgsGstreamerDevin
struct	AgsGstreamerDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsGstreamerDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsGstreamerDevin
```

Implemented Interfaces

AgsGstreamerDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/gstreamer/ags_gstreamer_devin.h>
```

Description

AgsGstreamerDevin represents a soundcard and supports input.

Functions

AGS_GSTREAMER_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_GSTREAMER_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsGstreamerDevin *) obj)->obj_mutex))
```

ags_gstreamer_devin_error_quark ()

```
GQuark
ags_gstreamer_devin_error_quark ();
```

ags_gstreamer_devin_test_flags ()

```
gboolean
ags_gstreamer_devin_test_flags (AgsGstreamerDevin *gstreamer_devin,
                                guint flags);
```

Test *flags* to be set on *gstreamer_devin* .

Parameters

gstreamer_devin	the AgsGstreamerDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.6.0

ags_gstreamer_devin_set_flags ()

```
void
ags_gstreamer_devin_set_flags (AgsGstreamerDevin *gstreamer_devin,
                               guint flags);
```

Enable a feature of *gstreamer_devin* .

Parameters

gstreamer_devin	the AgsGstreamerDevin	
flags	see AgsGstreamerDevinFlags	

Since: 3.6.0

ags_gstreamer_devin_unset_flags ()

```
void
ags_gstreamer_devin_unset_flags (AgsGstreamerDevin *gstreamer_devin,
                                 guint flags);
```

Disable a feature of *gstreamer_devin* .

Parameters

gstreamer_devin	the AgsGstreamerDevin	
flags	see AgsGstreamerDevinFlags	

Since: 3.6.0

ags_gstreamer_devin_switch_buffer_flag ()

```
void
ags_gstreamer_devin_switch_buffer_flag
                               (AgsGstreamerDevin *gstreamer_devin);
```

The buffer flag indicates the currently played buffer.

Parameters

gstreamer_devin	an AgsGstreamerDevin	

Since: 3.6.0

ags_gstreamer_devin_adjust_delay_and_attack ()

```
void
ags_gstreamer_devin_adjust_delay_and_attack
                                (AgsGstreamerDevin *gstreamer_devin);
```

Calculate delay and attack and reset it.

Parameters

gstreamer_devin	the AgsGstreamerDevin

Since: 3.6.0

ags_gstreamer_devin_realloc_buffer ()

```
void
ags_gstreamer_devin_realloc_buffer (AgsGstreamerDevin *gstreamer_devin);
```

Reallocate the internal audio buffer.

Parameters

gstreamer_devin	the AgsGstreamerDevin

Since: 3.6.0

ags_gstreamer_devin_new ()

```
AgsGstreamerDevin~*
ags_gstreamer_devin_new ();
```

Creates a new instance of AgsGstreamerDevin.

Returns

the new AgsGstreamerDevin

Since: 3.6.0

AGS_GSTREAMER_DEVIN()

```
#define AGS_GSTREAMER_DEVIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GSTREAMER_DEVIN, AgsGstreamerDevin))
```

AGS_GSTREAMER_DEVIN_CLASS()

```
#define AGS_GSTREAMER_DEVIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_GSTREAMER_DEVIN, AgsGstreamerDevin))
```

AGS_GSTREAMER_DEVIN_GET_CLASS()

```
#define AGS_GSTREAMER_DEVIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_GSTREAMER_DEVIN, AgsGstreamerDevinClass))
```

AGS_IS_GSTREAMER_DEVIN()

```
#define AGS_IS_GSTREAMER_DEVIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_GSTREAMER_DEVIN))
```

AGS_IS_GSTREAMER_DEVIN_CLASS()

```
#define AGS_IS_GSTREAMER_DEVIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_GSTREAMER_DEVIN))
```

ags_gstreamer_devin_flags_get_type ()

```
GType
ags_gstreamer_devin_flags_get_type ();
```

ags_gstreamer_devin_get_type ()

```
GType
ags_gstreamer_devin_get_type ();
```

Types and Values

AGS_GSTREAMER_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_GSTREAMER_DEVIN_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsGstreamerDevinFlags

Enum values to control the behavior or indicate internal state of AgsGstreamerDevin by enable/disable as flags.

Members

AGS_GSTREAMER_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_GSTREAMER_DEVIN_START_RECORD	capture start-ing

AGS_GSTREAMER_DEVIN_RECORD	do capture
AGS_GSTREAMER_DEVIN_SHUTDOWN	stop capture
AGS_GSTREAMER_DEVIN_NONBLOCKING	do non-blocking calls
AGS_GSTREAMER_DEVIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsGstreamerDevinAppBufferMode

Enum values to indicate internal state of AgsGstreamerDevin application buffer by setting mode.

Members

AGS_GSTREAMER_DEVIN_APP_BUFFER_0	ring-buffer 0
AGS_GSTREAMER_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_GSTREAMER_DEVIN_APP_BUFFER_2	ring-buffer 2
AGS_GSTREAMER_DEVIN_APP_BUFFER_3	ring-buffer 3
AGS_GSTREAMER_DEVIN_APP_BUFFER_4	ring-buffer 4
AGS_GSTREAMER_DEVIN_APP_BUFFER_5	ring-buffer 5
AGS_GSTREAMER_DEVIN_APP_BUFFER_6	ring-buffer 6
AGS_GSTREAMER_DEVIN_APP_BUFFER_7	ring-buffer 7

enum AgsGstreamerDevinSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_GSTREAMER_DEVIN_PASS_THROUGH	do not sync
AGS_GSTREAMER_DEVIN_INITIAL_CALLBACK	initial call-back
AGS_GSTREAMER_DEVIN_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_GSTREAMER_DEVIN_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_GSTREAMER_DEVIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_GSTREAMER_DEVIN_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_GSTREAMER_DEVIN_ERROR

```
#define AGS_GSTREAMER_DEVIN_ERROR (ags_gstreamer_devin_error_quark())
```

enum AgsGstreamerDevinError

Members

AGS_GSTREAMER_DEVIN_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_GSTREAMER_DEVIN

```
#define AGS_TYPE_GSTREAMER_DEVIN                (ags_gstreamer_devin_get_type())
```

struct AgsGstreamerDevin

```
struct AgsGstreamerDevin;
```

struct AgsGstreamerDevinClass

```
struct AgsGstreamerDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsGstreamerDevin

Flags: Read

Since: 3.6.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.6.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsGstreamerDevin

Flags: Read

Since: 3.6.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [1,44100]

Default value: 940

Since: 3.6.0

The "delay-factor" property

```
"delay-factor"               double
```

tact

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.6.0

The "device" property

```
"device"                     char~*
```

The gstreamer soundcard indentifier

Owner: AgsGstreamerDevin

Flags: Read / Write

Default value: "ags-gstreamer-devin-0"

Since: 3.6.0

The "dsp-channels" property

```
"dsp-channels"               guint
```

The dsp channel count

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.6.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.6.0

The "gstreamer-client" property

```
"gstreamer-client"          AgsGstreamerClient~*
```

The assigned AgsGstreamerClient

Owner: AgsGstreamerDevin

Flags: Read / Write

Since: 3.6.0

The "gstreamer-port" property

```
"gstreamer-port"            gpointer
```

The assigned AgsGstreamerPort

Owner: AgsGstreamerDevin

Flags: Read / Write

Since: 3.6.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.6.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsGstreamerDevin

Flags: Read / Write

Allowed values: [8000,96000]

Default value: 44100

Since: 3.6.0

8.25 AgsGstreamerDevout

AgsGstreamerDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_GSTREAMER_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_gstreamer_devout_error_quark ()
gboolean	ags_gstreamer_devout_test_flags ()
void	ags_gstreamer_devout_set_flags ()
void	ags_gstreamer_devout_unset_flags ()
void	ags_gstreamer_devout_switch_buffer_flag ()
void	ags_gstreamer_devout_adjust_delay_and_attack ()
void	ags_gstreamer_devout_realloc_buffer ()
AgsGstreamerDevout *	ags_gstreamer_devout_new ()
#define	AGS_GSTREAMER_DEVOUT()
#define	AGS_GSTREAMER_DEVOUT_CLASS()
#define	AGS_GSTREAMER_DEVOUT_GET_CLASS()
#define	AGS_IS_GSTREAMER_DEVOUT()
#define	AGS_IS_GSTREAMER_DEVOUT_CLASS()
GType	ags_gstreamer_devout_flags_get_type ()
GType	ags_gstreamer_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write

AgsGstreamerClient *	gstreamer-client	Read / Write
gpointer	gstreamer-port	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_GSTREAMER_DEVOUT_DEFAULT_APP_BUFFER_SIZE
enum	AgsGstreamerDevoutFlags
enum	AgsGstreamerDevoutAppBufferMode
enum	AgsGstreamerDevoutSyncFlags
#define	AGS_GSTREAMER_DEVOUT_ERROR
enum	AgsGstreamerDevoutError
#define	AGS_TYPE_GSTREAMER_DEVOUT
struct	AgsGstreamerDevout
struct	AgsGstreamerDevoutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsGstreamerDevoutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsGstreamerDevout
```

Implemented Interfaces

AgsGstreamerDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/gstreamer/ags_gstreamer_devout.h>
```

Description

AgsGstreamerDevout represents a soundcard and supports output.

Functions

AGS_GSTREAMER_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_GSTREAMER_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsGstreamerDevout *) obj)->obj_mutex) ↩
    )
```

ags_gstreamer_devout_error_quark ()

```
GQuark
ags_gstreamer_devout_error_quark ();
```

ags_gstreamer_devout_test_flags ()

```
gboolean
ags_gstreamer_devout_test_flags (AgsGstreamerDevout *gstreamer_devout,
                                 guint flags);
```

Test *flags* to be set on *gstreamer_devout* .

Parameters

gstreamer_devout	the AgsGstreamerDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.6.0

ags_gstreamer_devout_set_flags ()

```
void
ags_gstreamer_devout_set_flags (AgsGstreamerDevout *gstreamer_devout,
                                guint flags);
```

Enable a feature of *gstreamer_devout* .

Parameters

gstreamer_devout	the AgsGstreamerDevout	
flags	see AgsGstreamerDevoutFlags	

Since: 3.6.0

ags_gstreamer_devout_unset_flags ()

```
void
ags_gstreamer_devout_unset_flags (AgsGstreamerDevout *gstreamer_devout,
                                  guint flags);
```

Disable a feature of *gstreamer_devout* .

Parameters

gstreamer_devout	the AgsGstreamerDevout	
flags	see AgsGstreamerDevoutFlags	

Since: 3.6.0

ags_gstreamer_devout_switch_buffer_flag ()

```
void
ags_gstreamer_devout_switch_buffer_flag
                                (AgsGstreamerDevout *gstreamer_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

gstreamer_devout	an AgsGstreamerDevout

Since: 3.6.0

ags_gstreamer_devout_adjust_delay_and_attack ()

```
void
ags_gstreamer_devout_adjust_delay_and_attack
                                (AgsGstreamerDevout *gstreamer_devout);
```

Calculate delay and attack and reset it.

Parameters

gstreamer_devout	the AgsGstreamerDevout

Since: 3.6.0

ags_gstreamer_devout_realloc_buffer ()

```
void
ags_gstreamer_devout_realloc_buffer (AgsGstreamerDevout *gstreamer_devout);
```

Reallocate the internal audio buffer.

Parameters

gstreamer_devout	the AgsGstreamerDevout

Since: 3.6.0

ags_gstreamer_devout_new ()

```
AgsGstreamerDevout~*
ags_gstreamer_devout_new ();
```

Creates a new instance of AgsGstreamerDevout.

Returns

the new AgsGstreamerDevout

Since: 3.6.0

AGS_GSTREAMER_DEVOUT()

```
#define AGS_GSTREAMER_DEVOUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GSTREAMER_DEVOUT, AgsGstreamerDevout))
```

AGS_GSTREAMER_DEVOUT_CLASS()

```
#define AGS_GSTREAMER_DEVOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_GSTREAMER_DEVOUT, AgsGstreamerDevout))
```

AGS_GSTREAMER_DEVOUT_GET_CLASS()

```
#define AGS_GSTREAMER_DEVOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_GSTREAMER_DEVOUT, AgsGstreamerDevoutClass))
```

AGS_IS_GSTREAMER_DEVOUT()

```
#define AGS_IS_GSTREAMER_DEVOUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_GSTREAMER_DEVOUT))
```

AGS_IS_GSTREAMER_DEVOUT_CLASS()

```
#define AGS_IS_GSTREAMER_DEVOUT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_GSTREAMER_DEVOUT))
```

ags_gstreamer_devout_flags_get_type ()

```
GType
ags_gstreamer_devout_flags_get_type ();
```

ags_gstreamer_devout_get_type ()

```
GType
ags_gstreamer_devout_get_type ();
```

Types and Values

AGS_GSTREAMER_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_GSTREAMER_DEVOUT_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsGstreamerDevoutFlags

Enum values to control the behavior or indicate internal state of AgsGstreamerDevout by enable/disable as flags.

Members

AGS_GSTREAMER_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_GSTREAMER_DEVOUT_START_PLAY	playback start-ing
AGS_GSTREAMER_DEVOUT_PLAY	do play-back
AGS_GSTREAMER_DEVOUT_SHUTDOWN	stop play-back
AGS_GSTREAMER_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_GSTREAMER_DEVOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsGstreamerDevoutAppBufferMode

Enum values to indicate internal state of AgsGstreamerDevout application buffer by setting mode.

Members

AGS_GSTREAMER_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_GSTREAMER_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_GSTREAMER_DEVOUT_APP_BUFFER_2	ring-buffer 2

AGS_GSTREAMER_DEVOUT_APP_BUFFER_3	ring-buffer 3
AGS_GSTREAMER_DEVOUT_APP_BUFFER_4	ring-buffer 4
AGS_GSTREAMER_DEVOUT_APP_BUFFER_5	ring-buffer 5
AGS_GSTREAMER_DEVOUT_APP_BUFFER_6	ring-buffer 6
AGS_GSTREAMER_DEVOUT_APP_BUFFER_7	ring-buffer 7

enum AgsGstreamerDevoutSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_GSTREAMER_DEVOUT_PASS_THROUGH	do not sync
AGS_GSTREAMER_DEVOUT_INITIAL_CALLBACK	initial call-back
AGS_GSTREAMER_DEVOUT_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_GSTREAMER_DEVOUT_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_GSTREAMER_DEVOUT_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock

AGS_GSTREAMER_DEVOUT_CALLBACK_FINISH_DONE	sync done, client conditional lock

AGS_GSTREAMER_DEVOUT_ERROR

```
#define AGS_GSTREAMER_DEVOUT_ERROR (ags_gstreamer_devout_error_quark())
```

enum AgsGstreamerDevoutError

Members

AGS_GSTREAMER_DEVOUT_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_GSTREAMER_DEVOUT

```
#define AGS_TYPE_GSTREAMER_DEVOUT                (ags_gstreamer_devout_get_type())
```

struct AgsGstreamerDevout

```
struct AgsGstreamerDevout;
```

struct AgsGstreamerDevoutClass

```
struct AgsGstreamerDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsGstreamerDevout

Flags: Read

Since: 3.6.0

The "bpm" property

```
"bpm"                    double
```

Beats per minute

Owner: AgsGstreamerDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.6.0

The "buffer" property

```
"buffer"                 gpointer
```

The buffer

Owner: AgsGstreamerDevout

Flags: Read

Since: 3.6.0

The "buffer-size" property

```
"buffer-size"            guint
```

The buffer size

Owner: AgsGstreamerDevout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.6.0

The "delay-factor" property

```
"delay-factor"           double
```

tact

Owner: AgsGstreamerDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.6.0

The "device" property

```
"device"                    char~*
```

The gstreamer soundcard indentifier

Owner: AgsGstreamerDevout

Flags: Read / Write

Default value: "ags-gstreamer-devout-0"

Since: 3.6.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsGstreamerDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.6.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsGstreamerDevout

Flags: Read / Write

Default value: 16

Since: 3.6.0

The "gstreamer-client" property

```
"gstreamer-client"          AgsGstreamerClient~*
```

The assigned AgsGstreamerClient

Owner: AgsGstreamerDevout

Flags: Read / Write

Since: 3.6.0

The "gstreamer-port" property

```
"gstreamer-port"            gpointer
```

The assigned AgsGstreamerPort

Owner: AgsGstreamerDevout

Flags: Read / Write

Since: 3.6.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsGstreamerDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.6.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsGstreamerDevout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.6.0

8.26 AgsCoreAudioServer

AgsCoreAudioServer — core audio instance

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CORE_AUDIO_SERVER_GET_OBJ_MUTEX()
gboolean	ags_core_audio_server_test_flags ()
void	ags_core_audio_server_set_flags ()
void	ags_core_audio_server_unset_flags ()
GList *	ags_core_audio_server_find_url ()
GObject *	ags_core_audio_server_find_client ()
GObject *	ags_core_audio_server_find_port ()
void	ags_core_audio_server_add_client ()
void	ags_core_audio_server_remove_client ()
void	ags_core_audio_server_connect_client ()
void	ags_core_audio_server_start_poll ()
AgsCoreAudioServer *	ags_core_audio_server_new ()
#define	AGS_CORE_AUDIO_SERVER()
#define	AGS_CORE_AUDIO_SERVER_CLASS()

#define	AGS_CORE_AUDIO_SERVER_GET_CLASS()
#define	AGS_IS_CORE_AUDIO_SERVER()
#define	AGS_IS_CORE_AUDIO_SERVER_CLASS()
GType	ags_core_audio_server_get_type ()

Properties

gpointer	core-audio-client	Read / Write
AgsCoreAudioClient *	default-core-audio-client	Read / Write
GObject *	default-soundcard	Read / Write
char *	url	Read / Write

Types and Values

#define	AGS_TYPE_CORE_AUDIO_SERVER
struct	AgsCoreAudioServer
struct	AgsCoreAudioServerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsCoreAudioServer
```

Implemented Interfaces

AgsCoreAudioServer implements AgsConnectable and AgsSoundServer.

Includes

```
#include <ags/audio/core-audio/ags_core_audio_server.h>
```

Description

The AgsCoreAudioServer is an object to represent a running core audio instance.

Functions

AGS_CORE_AUDIO_SERVER_GET_OBJ_MUTEX()

```
#define AGS_CORE_AUDIO_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsCoreAudioServer *) obj)->obj_mutex ↩
    ))
```

ags_core_audio_server_test_flags ()

```
gboolean
ags_core_audio_server_test_flags (AgsCoreAudioServer *core_audio_server,
                                  guint flags);
```

Test *flags* to be set on *core_audio_server* .

Parameters

core_audio_server	the AgsCoreAudioServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_core_audio_server_set_flags ()

```
void
ags_core_audio_server_set_flags (AgsCoreAudioServer *core_audio_server,
                                 guint flags);
```

Enable a feature of *core_audio_server* .

Parameters

core_audio_server	the AgsCoreAudioServer	
flags	see AgsCoreAudioServerFlags	

Since: 3.0.0

ags_core_audio_server_unset_flags ()

```
void
ags_core_audio_server_unset_flags (AgsCoreAudioServer *core_audio_server,
                                   guint flags);
```

Disable a feature of *core_audio_server* .

Parameters

core_audio_server	the AgsCoreAudioServer	
flags	see AgsCoreAudioServerFlags	

Since: 3.0.0

ags_core_audio_server_find_url ()

```
GList~*
ags_core_audio_server_find_url (GList *core_audio_server,
                                gchar *url);
```

Find AgsCoreAudioServer by url.

Parameters

core_audio_server	the GList containing AgsCoreAudioServer.	*[element-type AgsAudio.CoreAudioServer][transfer none]*
url	the url to find	

Returns

the next matching GList containing a AgsCoreAudioServer matching `url` or NULL.

[element-type AgsAudio.CoreAudioServer][transfer none]

Since: 3.0.0

ags_core_audio_server_find_client ()

```
GObject~*
ags_core_audio_server_find_client (AgsCoreAudioServer *core_audio_server,
                                   gchar *client_uuid);
```

Find AgsCoreAudioClient by uuid.

Parameters

core_audio_server	the AgsCoreAudioServer	
client_uuid	the uuid to find	

Returns

the AgsCoreAudioClient found or NULL.

[transfer none]

Since: 3.0.0

ags_core_audio_server_find_port ()

```
GObject~*
ags_core_audio_server_find_port (AgsCoreAudioServer *core_audio_server,
                                 gchar *port_uuid);
```

Find AgsCoreAudioPort by uuid.

Parameters

core_audio_server	the AgsCoreAudioServer	
port_uuid	the uuid to find	

Returns

the AgsCoreAudioPort found or NULL.

[transfer none]

Since: 3.0.0

ags_core_audio_server_add_client ()

```
void
ags_core_audio_server_add_client (AgsCoreAudioServer *core_audio_server,
                                  GObject *core_audio_client);
```

Add *core_audio_client* to *core_audio_server*

Parameters

core_audio_server	the AgsCoreAudioServer	
core_audio_client	the AgsCoreAudioClient to add	

Since: 3.0.0

ags_core_audio_server_remove_client ()

```
void
ags_core_audio_server_remove_client (AgsCoreAudioServer *core_audio_server,
                                     GObject *core_audio_client);
```

Remove *core_audio_client* to *core_audio_server*

Parameters

core_audio_server	the AgsCoreAudioServer	
core_audio_client	the AgsCoreAudioClient to remove	

Since: 3.0.0

ags_core_audio_server_connect_client ()

```
void
ags_core_audio_server_connect_client (AgsCoreAudioServer *core_audio_server);
```

Connect all clients.

Parameters

core_audio_server	the AgsCoreAudioServer	

Since: 3.0.0

ags_core_audio_server_start_poll ()

```
void
ags_core_audio_server_start_poll (AgsCoreAudioServer *core_audio_server);
```

ags_core_audio_server_new ()

```
AgsCoreAudioServer~*
ags_core_audio_server_new (gchar *url);
```

Create a new instance of AgsCoreAudioServer.

Parameters

url	the URL as string

Returns

the new AgsCoreAudioServer

Since: 3.0.0

AGS_CORE_AUDIO_SERVER()

```
#define AGS_CORE_AUDIO_SERVER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CORE_AUDIO_SERVER, AgsCoreAudioServer))
```

AGS_CORE_AUDIO_SERVER_CLASS()

```
#define AGS_CORE_AUDIO_SERVER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CORE_AUDIO_SERVER, AgsCoreAudioServer))
```

AGS_CORE_AUDIO_SERVER_GET_CLASS()

```
#define AGS_CORE_AUDIO_SERVER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CORE_AUDIO_SERVER, AgsCoreAudioServerClass))
```

AGS_IS_CORE_AUDIO_SERVER()

```
#define AGS_IS_CORE_AUDIO_SERVER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CORE_AUDIO_SERVER))
```

AGS_IS_CORE_AUDIO_SERVER_CLASS()

```
#define AGS_IS_CORE_AUDIO_SERVER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CORE_AUDIO_SERVER))
```

ags_core_audio_server_get_type ()

```
GType
ags_core_audio_server_get_type ();
```

Types and Values

AGS_TYPE_CORE_AUDIO_SERVER

```
#define AGS_TYPE_CORE_AUDIO_SERVER                (ags_core_audio_server_get_type())
```

struct AgsCoreAudioServer

```
struct AgsCoreAudioServer;
```

struct AgsCoreAudioServerClass

```
struct AgsCoreAudioServerClass {
  GObjectClass gobject;
};
```

Property Details

The "core-audio-client" property

```
  "core-audio-client"         gpointer
```

The core audio client list.

[transfer full]

Owner: AgsCoreAudioServer

Flags: Read / Write

Since: 3.0.0

The "default-core-audio-client" property

```
  "default-core-audio-client" AgsCoreAudioClient~*
```

The default core audio client.

Owner: AgsCoreAudioServer

Flags: Read / Write

Since: 3.0.0

The "default-soundcard" property

```
  "default-soundcard"         GObject~*
```

The default soundcard.

Owner: AgsCoreAudioServer

Flags: Read / Write

Since: 3.0.0

The "url" property

```
"url"                    char~*
```

The assigned URL.

Owner: AgsCoreAudioServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

8.27 AgsCoreAudioClient

AgsCoreAudioClient — core audio connection

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CORE_AUDIO_CLIENT_GET_OBJ_MUTEX()
gboolean	ags_core_audio_client_test_flags ()
void	ags_core_audio_client_set_flags ()
void	ags_core_audio_client_unset_flags ()
GList *	ags_core_audio_client_find_uuid ()
GList *	ags_core_audio_client_find ()
void	ags_core_audio_client_open ()
void	ags_core_audio_client_add_device ()
void	ags_core_audio_client_remove_device ()
void	ags_core_audio_client_add_port ()
void	ags_core_audio_client_remove_port ()
void	ags_core_audio_client_activate ()
void	ags_core_audio_client_deactivate ()
AgsCoreAudioClient *	ags_core_audio_client_new ()
#define	AGS_CORE_AUDIO_CLIENT()
#define	AGS_CORE_AUDIO_CLIENT_CLASS()
#define	AGS_CORE_AUDIO_CLIENT_GET_CLASS()
#define	AGS_IS_CORE_AUDIO_CLIENT()
#define	AGS_IS_CORE_AUDIO_CLIENT_CLASS()
GType	ags_core_audio_client_flags_get_type ()
GType	ags_core_audio_client_get_type ()

Properties

char *	client-name	Read / Write
AgsCoreAudioServer *	core-audio-server	Read / Write
gpointer	device	Read / Write
gpointer	port	Read / Write

Types and Values

enum	AgsCoreAudioClientFlags
#define	AGS_TYPE_CORE_AUDIO_CLIENT
struct	AgsCoreAudioClient
struct	AgsCoreAudioClientClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsCoreAudioClientFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsCoreAudioClient
```

Implemented Interfaces

AgsCoreAudioClient implements AgsConnectable.

Includes

```
#include <ags/audio/core-audio/ags_core_audio_client.h>
```

Description

The AgsCoreAudioClient communicates with a core audio instance.

Functions

AGS_CORE_AUDIO_CLIENT_GET_OBJ_MUTEX()

```
#define AGS_CORE_AUDIO_CLIENT_GET_OBJ_MUTEX(obj) (&(((AgsCoreAudioClient *) obj)->obj_mutex ↩
    ))
```

ags_core_audio_client_test_flags ()

```
gboolean
ags_core_audio_client_test_flags (AgsCoreAudioClient *core_audio_client,
                                  guint flags);
```

Test *flags* to be set on *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_core_audio_client_set_flags ()

```
void
ags_core_audio_client_set_flags (AgsCoreAudioClient *core_audio_client,
                                 guint flags);
```

Enable a feature of *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
flags	see AgsCoreAudioClientFlags	

Since: 3.0.0

ags_core_audio_client_unset_flags ()

```
void
ags_core_audio_client_unset_flags (AgsCoreAudioClient *core_audio_client,
                                   guint flags);
```

Disable a feature of *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
flags	see AgsCoreAudioClientFlags	

Since: 3.0.0

ags_core_audio_client_find_uuid ()

```
GList~*
ags_core_audio_client_find_uuid (GList *core_audio_client,
                                 gchar *client_uuid);
```

Finds next match of *client_uuid* in *core_audio_client* .

Parameters

core_audio_client	the GList containing AgsCoreAudioClient.	*[element-type AgsAudio.CoreAudioClient][transfer none]*
client_uuid	the client uuid to find	

Returns

a GList or NULL.

[element-type AgsAudio.CoreAudioClient][transfer none]

Since: 3.0.0

ags_core_audio_client_find ()

```
GList~*
ags_core_audio_client_find (GList *core_audio_client,
                            gchar *client_name);
```

Finds next match of *client_name* in *core_audio_client* .

Parameters

core_audio_client	the GList containing AgsCoreAudioClient.	*[element-type AgsAudio.CoreAudioClient][transfer none]*
client_name	the client name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.CoreAudioClient][transfer none]

Since: 3.0.0

ags_core_audio_client_open ()

```
void
ags_core_audio_client_open (AgsCoreAudioClient *core_audio_client,
                            gchar *client_name);
```

Open the core audio client's connection and read uuid.

Parameters

core_audio_client	the AgsCoreAudioClient	
client_name	the client's name	

Since: 3.0.0

ags_core_audio_client_add_device ()

```
void
ags_core_audio_client_add_device (AgsCoreAudioClient *core_audio_client,
                                  GObject *core_audio_device);
```

Add *core_audio_device* to *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
core_audio_device	an AgsCoreAudioDevout or AgsCoreAudioDevin	

Since: 3.0.0

ags_core_audio_client_remove_device ()

```
void
ags_core_audio_client_remove_device (AgsCoreAudioClient *core_audio_client,
                                     GObject *core_audio_device);
```

Remove *core_audio_device* from *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
core_audio_device	an AgsCoreAudioDevout or AgsCoreAudioDevin	

Since: 3.0.0

ags_core_audio_client_add_port ()

```
void
ags_core_audio_client_add_port (AgsCoreAudioClient *core_audio_client,
                                GObject *core_audio_port);
```

Add *core_audio_port* to *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
core_audio_port	an AgsCoreAudioPort	

Since: 3.0.0

ags_core_audio_client_remove_port ()

```
void
ags_core_audio_client_remove_port (AgsCoreAudioClient *core_audio_client,
                                   GObject *core_audio_port);
```

Remove *core_audio_port* from *core_audio_client* .

Parameters

core_audio_client	the AgsCoreAudioClient	
core_audio_port	an AgsCoreAudioPort	

Since: 3.0.0

ags_core_audio_client_activate ()

```
void
ags_core_audio_client_activate (AgsCoreAudioClient *core_audio_client);
```

Activate client.

Parameters

core_audio_client	the AgsCoreAudioClient

Since: 3.0.0

ags_core_audio_client_deactivate ()

```
void
ags_core_audio_client_deactivate (AgsCoreAudioClient *core_audio_client);
```

Deactivate client.

Parameters

core_audio_client	the AgsCoreAudioClient

Since: 3.0.0

ags_core_audio_client_new ()

```
AgsCoreAudioClient~*
ags_core_audio_client_new (GObject *core_audio_server);
```

Create a new instance of AgsCoreAudioClient.

Parameters

core_audio_server	the assigned AgsCoreAudioServer

Returns

the new AgsCoreAudioClient

Since: 3.0.0

AGS_CORE_AUDIO_CLIENT()

```
#define AGS_CORE_AUDIO_CLIENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CORE_AUDIO_CLIENT, AgsCoreAudioClient))
```

AGS_CORE_AUDIO_CLIENT_CLASS()

```
#define AGS_CORE_AUDIO_CLIENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CORE_AUDIO_CLIENT, AgsCoreAudioClient))
```

AGS_CORE_AUDIO_CLIENT_GET_CLASS()

```
#define AGS_CORE_AUDIO_CLIENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CORE_AUDIO_CLIENT, AgsCoreAudioClientClass))
```

AGS_IS_CORE_AUDIO_CLIENT()

```
#define AGS_IS_CORE_AUDIO_CLIENT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CORE_AUDIO_CLIENT))
```

AGS_IS_CORE_AUDIO_CLIENT_CLASS()

```
#define AGS_IS_CORE_AUDIO_CLIENT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CORE_AUDIO_CLIENT))
```

ags_core_audio_client_flags_get_type ()

```
GType
ags_core_audio_client_flags_get_type ();
```

ags_core_audio_client_get_type ()

```
GType
ags_core_audio_client_get_type ();
```

Types and Values

enum AgsCoreAudioClientFlags

Enum values to control the behavior or indicate internal state of AgsCoreAudioClient by enable/disable as flags.

Members

AGS_CORE_AUDIO_CLIENT_ACTIVATED	the client was activated
AGS_CORE_AUDIO_CLIENT_READY	the client is ready

AGS_TYPE_CORE_AUDIO_CLIENT

```
#define AGS_TYPE_CORE_AUDIO_CLIENT                (ags_core_audio_client_get_type())
```

struct AgsCoreAudioClient

```
struct AgsCoreAudioClient;
```

struct AgsCoreAudioClientClass

```
struct AgsCoreAudioClientClass {
  GObjectClass gobject;
};
```

Property Details

The "client-name" property

```
  "client-name"              char~*
```

The core-audio client name.

Owner: AgsCoreAudioClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "core-audio-server" property

```
  "core-audio-server"        AgsCoreAudioServer~*
```

The assigned AgsCoreAudioServer.

Owner: AgsCoreAudioClient

Flags: Read / Write

Since: 3.0.0

The "device" property

```
  "device"                   gpointer
```

The assigned devices.

[transfer full]

Owner: AgsCoreAudioClient

Flags: Read / Write

Since: 3.0.0

The "port" property

```
"port"                    gpointer
```

The assigned ports.

[transfer full]

Owner: AgsCoreAudioClient

Flags: Read / Write

Since: 3.0.0

8.28 AgsCoreAudioPort

AgsCoreAudioPort — core audio resource.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CORE_AUDIO_PORT_GET_OBJ_MUTEX()
gboolean	ags_core_audio_port_test_flags ()
void	ags_core_audio_port_set_flags ()
void	ags_core_audio_port_unset_flags ()
GList *	ags_core_audio_port_find ()
void	ags_core_audio_port_register ()
void	ags_core_audio_port_unregister ()
void	ags_core_audio_port_set_format ()
void	ags_core_audio_port_set_samplerate ()
void	ags_core_audio_port_set_pcm_channels ()
void	ags_core_audio_port_set_buffer_size ()
void	ags_core_audio_port_set_cache_buffer_size ()
guint	ags_core_audio_port_get_latency ()
AgsCoreAudioPort *	ags_core_audio_port_new ()
#define	AGS_CORE_AUDIO_PORT()
#define	AGS_CORE_AUDIO_PORT_CLASS()
#define	AGS_CORE_AUDIO_PORT_GET_CLASS()
#define	AGS_IS_CORE_AUDIO_PORT()
#define	AGS_IS_CORE_AUDIO_PORT_CLASS()
GType	ags_core_audio_port_flags_get_type ()
GType	ags_core_audio_port_get_type ()

Properties

AgsCoreAudioClient *	core-audio-client	Read / Write
GObject *	core-audio-device	Read / Write
char *	port-name	Read / Write

Types and Values

#define	AGS_CORE_AUDIO_PORT_DEFAULT_CACHE_BUFFER_SIZE
enum	AgsCoreAudioPortFlags
#define	AGS_TYPE_CORE_AUDIO_PORT
struct	AgsCoreAudioPort
struct	AgsCoreAudioPortClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsCoreAudioPortFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsCoreAudioPort
```

Implemented Interfaces

AgsCoreAudioPort implements AgsConnectable.

Includes

```
#include <ags/audio/core-audio/ags_core_audio_port.h>
```

Description

The AgsCoreAudioPort represents either a core audio sequencer or soundcard to communicate with.

Functions

AGS_CORE_AUDIO_PORT_GET_OBJ_MUTEX()

```
#define AGS_CORE_AUDIO_PORT_GET_OBJ_MUTEX(obj) (&(((AgsCoreAudioPort *) obj)->obj_mutex))
```

ags_core_audio_port_test_flags ()

```
gboolean
ags_core_audio_port_test_flags (AgsCoreAudioPort *core_audio_port,
                                guint flags);
```

Test *flags* to be set on *core_audio_port* .

Parameters

core_audio_port	the AgsCoreAudioPort	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_core_audio_port_set_flags ()

```
void
ags_core_audio_port_set_flags (AgsCoreAudioPort *core_audio_port,
                               guint flags);
```

Enable a feature of *core_audio_port* .

Parameters

core_audio_port	the AgsCoreAudioPort	
flags	see AgsCoreAudioPortFlags	

Since: 3.0.0

ags_core_audio_port_unset_flags ()

```
void
ags_core_audio_port_unset_flags (AgsCoreAudioPort *core_audio_port,
                                 guint flags);
```

Disable a feature of *core_audio_port* .

Parameters

core_audio_port	the AgsCoreAudioPort	
flags	see AgsCoreAudioPortFlags	

Since: 3.0.0

ags_core_audio_port_find ()

```
GList~*
ags_core_audio_port_find (GList *core_audio_port,
                          gchar *port_name);
```

Finds next match of *port_name* in *core_audio_port* .

Parameters

core_audio_port	the GList containing AgsCoreAudioPort.	*[element-type AgsAudio.CoreAudioPort][transfer none]*
port_name	the port name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.CoreAudioPort][transfer none]

Since: 3.0.0

ags_core_audio_port_register ()

```
void
ags_core_audio_port_register (AgsCoreAudioPort *core_audio_port,
                              gchar *port_name,
                              gboolean is_audio,
                              gboolean is_midi,
                              gboolean is_output);
```

Register a new core audio port and read uuid. Creates a new AgsSequencer or AgsSoundcard object.

Parameters

core_audio_port	the AgsCoreAudioPort	
port_name	the name as string	
is_audio	if TRUE interpreted as audio port	
is_midi	if TRUE interpreted as midi port	
is_output	if TRUE port is acting as output, otherwise as input	

Since: 3.0.0

ags_core_audio_port_unregister ()

```
void
ags_core_audio_port_unregister (AgsCoreAudioPort *core_audio_port);
```

ags_core_audio_port_set_format ()

```
void
ags_core_audio_port_set_format (AgsCoreAudioPort *core_audio_port,
                                guint format);
```

ags_core_audio_port_set_samplerate ()

```
void
ags_core_audio_port_set_samplerate (AgsCoreAudioPort *core_audio_port,
                                    guint samplerate);
```

ags_core_audio_port_set_pcm_channels ()

```
void
ags_core_audio_port_set_pcm_channels (AgsCoreAudioPort *core_audio_port,
                                      guint pcm_channels);
```

ags_core_audio_port_set_buffer_size ()

```
void
ags_core_audio_port_set_buffer_size (AgsCoreAudioPort *core_audio_port,
                                     guint buffer_size);
```

ags_core_audio_port_set_cache_buffer_size ()

```
void
ags_core_audio_port_set_cache_buffer_size
                                (AgsCoreAudioPort *core_audio_port,
                                 guint cache_buffer_size);
```

ags_core_audio_port_get_latency ()

```
guint
ags_core_audio_port_get_latency (AgsCoreAudioPort *core_audio_port);
```

Gets latency.

Parameters

core_audio_port	the AgsCoreAudioPort

Since: 3.0.0

ags_core_audio_port_new ()

```
AgsCoreAudioPort~*
ags_core_audio_port_new (GObject *core_audio_client);
```

Create a new instance of AgsCoreAudioPort.

Parameters

core_audio_client	the AgsCoreAudioClient assigned to

Returns

the new AgsCoreAudioPort

Since: 3.0.0

AGS_CORE_AUDIO_PORT()

```
#define AGS_CORE_AUDIO_PORT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CORE_AUDIO_PORT, AgsCoreAudioPort))
```

AGS_CORE_AUDIO_PORT_CLASS()

```
#define AGS_CORE_AUDIO_PORT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CORE_AUDIO_PORT, AgsCoreAudioPort))
```

AGS_CORE_AUDIO_PORT_GET_CLASS()

```
#define AGS_CORE_AUDIO_PORT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CORE_AUDIO_PORT, AgsCoreAudioPortClass))
```

AGS_IS_CORE_AUDIO_PORT()

```
#define AGS_IS_CORE_AUDIO_PORT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CORE_AUDIO_PORT))
```

AGS_IS_CORE_AUDIO_PORT_CLASS()

```
#define AGS_IS_CORE_AUDIO_PORT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CORE_AUDIO_PORT))
```

ags_core_audio_port_flags_get_type ()

```
GType
ags_core_audio_port_flags_get_type ();
```

ags_core_audio_port_get_type ()

```
GType
ags_core_audio_port_get_type ();
```

Types and Values

AGS_CORE_AUDIO_PORT_DEFAULT_CACHE_BUFFER_SIZE

```
#define AGS_CORE_AUDIO_PORT_DEFAULT_CACHE_BUFFER_SIZE (4096)
```

enum AgsCoreAudioPortFlags

Enum values to control the behavior or indicate internal state of AgsCoreAudioPort by enable/disable as flags.

Members

AGS_CORE_AUDIO_PORT_REGISTERED	the port was registered

AGS_CORE_AUDIO_PORT_IS_AUDIO	the port provides audio data
AGS_CORE_AUDIO_PORT_IS_MIDI	the port provides midi data
AGS_CORE_AUDIO_PORT_IS_OUTPUT	the port does output
AGS_CORE_AUDIO_PORT_IS_INPUT	the port does input

AGS_TYPE_CORE_AUDIO_PORT

```
#define AGS_TYPE_CORE_AUDIO_PORT                (ags_core_audio_port_get_type())
```

struct AgsCoreAudioPort

```
struct AgsCoreAudioPort;
```

struct AgsCoreAudioPortClass

```
struct AgsCoreAudioPortClass {
  GObjectClass gobject;
};
```

Property Details

The "core-audio-client" property

```
  "core-audio-client"        AgsCoreAudioClient~*
```

The assigned AgsCoreAudioClient.

Owner: AgsCoreAudioPort

Flags: Read / Write

Since: 3.0.0

The "core-audio-device" property

```
"core-audio-device"        GObject~*
```

The assigned AgsCoreAudioDevout.

Owner: AgsCoreAudioPort

Flags: Read / Write

Since: 3.0.0

The "port-name" property

```
"port-name"                char~*
```

The core audio soundcard indentifier

Owner: AgsCoreAudioPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

8.29 AgsCoreAudioDevin

AgsCoreAudioDevin — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CORE_AUDIO_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_core_audio_devin_error_quark ()
gboolean	ags_core_audio_devin_test_flags ()
void	ags_core_audio_devin_set_flags ()
void	ags_core_audio_devin_unset_flags ()
void	ags_core_audio_devin_switch_buffer_flag ()
void	ags_core_audio_devin_adjust_delay_and_attack ()
void	ags_core_audio_devin_realloc_buffer ()
AgsCoreAudioDevin *	ags_core_audio_devin_new ()
#define	AGS_CORE_AUDIO_DEVIN()
#define	AGS_CORE_AUDIO_DEVIN_CLASS()
#define	AGS_CORE_AUDIO_DEVIN_GET_CLASS()
#define	AGS_IS_CORE_AUDIO_DEVIN()
#define	AGS_IS_CORE_AUDIO_DEVIN_CLASS()
GType	ags_core_audio_devin_flags_get_type ()
GType	ags_core_audio_devin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
AgsCoreAudioClient *	core-audio-client	Read / Write
gpointer	core-audio-port	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_CORE_AUDIO_DEVIN_DEFAULT_APP_BUFFER_SIZE
enum	AgsCoreAudioDevinFlags
enum	AgsCore_AudioDevinAppBufferMode
enum	AgsCoreAudioDevinSyncFlags
#define	AGS_CORE_AUDIO_DEVIN_ERROR
enum	AgsCoreAudioDevinError
#define	AGS_TYPE_CORE_AUDIO_DEVIN
struct	AgsCoreAudioDevin
struct	AgsCoreAudioDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsCoreAudioDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsCoreAudioDevin
```

Implemented Interfaces

AgsCoreAudioDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/core-audio/ags_core_audio_devin.h>
```

Description

AgsCoreAudioDevin represents a soundcard and supports output.

Functions

AGS_CORE_AUDIO_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_CORE_AUDIO_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsCoreAudioDevin *) obj)->obj_mutex))
```

ags_core_audio_devin_error_quark ()

```
GQuark
ags_core_audio_devin_error_quark ();
```

ags_core_audio_devin_test_flags ()

```
gboolean
ags_core_audio_devin_test_flags (AgsCoreAudioDevin *core_audio_devin,
                                 guint flags);
```

Test *flags* to be set on *core_audio_devin* .

Parameters

core_audio_devin	the AgsCoreAudioDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_core_audio_devin_set_flags ()

```
void
ags_core_audio_devin_set_flags (AgsCoreAudioDevin *core_audio_devin,
                                guint flags);
```

Enable a feature of *core_audio_devin* .

Parameters

core_audio_devin	the AgsCoreAudioDevin	
flags	see AgsCoreAudioDevinFlags	

Since: 3.0.0

ags_core_audio_devin_unset_flags ()

```
void
ags_core_audio_devin_unset_flags (AgsCoreAudioDevin *core_audio_devin,
                                  guint flags);
```

Disable a feature of *core_audio_devin* .

Parameters

core_audio_devin	the AgsCoreAudioDevin	
flags	see AgsCoreAudioDevinFlags	

Since: 3.0.0

ags_core_audio_devin_switch_buffer_flag ()

```
void
ags_core_audio_devin_switch_buffer_flag
                                (AgsCoreAudioDevin *core_audio_devin);
```

The buffer flag indicates the currently played buffer.

Parameters

core_audio_devin	an AgsCoreAudioDevin	

Since: 3.0.0

ags_core_audio_devin_adjust_delay_and_attack ()

```
void
ags_core_audio_devin_adjust_delay_and_attack
                                (AgsCoreAudioDevin *core_audio_devin);
```

Calculate delay and attack and reset it.

Parameters

core_audio_devin	the AgsCoreAudioDevin	

Since: 3.0.0

ags_core_audio_devin_realloc_buffer ()

```
void
ags_core_audio_devin_realloc_buffer (AgsCoreAudioDevin *core_audio_devin);
```

Reallocate the internal audio buffer.

Parameters

core_audio_devin	the AgsCoreAudioDevin	

Since: 3.0.0

ags_core_audio_devin_new ()

```
AgsCoreAudioDevin~*
ags_core_audio_devin_new ();
```

Creates a new instance of AgsCoreAudioDevin.

Returns

a new AgsCoreAudioDevin

Since: 3.0.0

AGS_CORE_AUDIO_DEVIN()

```
#define AGS_CORE_AUDIO_DEVIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CORE_AUDIO_DEVIN, AgsCoreAudioDevin))
```

AGS_CORE_AUDIO_DEVIN_CLASS()

```
#define AGS_CORE_AUDIO_DEVIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CORE_AUDIO_DEVIN, AgsCoreAudioDevin))
```

AGS_CORE_AUDIO_DEVIN_GET_CLASS()

```
#define AGS_CORE_AUDIO_DEVIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CORE_AUDIO_DEVIN, AgsCoreAudioDevinClass))
```

AGS_IS_CORE_AUDIO_DEVIN()

```
#define AGS_IS_CORE_AUDIO_DEVIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CORE_AUDIO_DEVIN))
```

AGS_IS_CORE_AUDIO_DEVIN_CLASS()

```
#define AGS_IS_CORE_AUDIO_DEVIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CORE_AUDIO_DEVIN))
```

ags_core_audio_devin_flags_get_type ()

```
GType
ags_core_audio_devin_flags_get_type ();
```

ags_core_audio_devin_get_type ()

```
GType
ags_core_audio_devin_get_type ();
```

Types and Values

AGS_CORE_AUDIO_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_CORE_AUDIO_DEVIN_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsCoreAudioDevinFlags

Enum values to control the behavior or indicate internal state of AgsCoreAudioDevin by enable/disable as flags.

Members

AGS_CORE_AUDIO_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_CORE_AUDIO_DEVIN_START_RECORD	capture start-ing
AGS_CORE_AUDIO_DEVIN_RECORD	do cap-ture
AGS_CORE_AUDIO_DEVIN_SHUTDOWN	stop cap-ture
AGS_CORE_AUDIO_DEVIN_NONBLOCKING	do non-blocking calls
AGS_CORE_AUDIO_DEVIN_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsCore_AudioDevinAppBufferMode

Members

AGS_CORE_AUDIO_DEVIN_APP_BUFFER_0	
AGS_CORE_AUDIO_DEVIN_APP_BUFFER_1	
AGS_CORE_AUDIO_DEVIN_APP_BUFFER_2	
AGS_CORE_AUDIO_DEVIN_APP_BUFFER_3	
AGS_CORE_AUDIO_DEVIN_APP_BUFFER_4	
AGS_CORE_AUDIO_DEVIN_APP_BUFFER_5	

AGS_CORE_AUDIO_DEVIN_APP_BUFFER_6	
AGS_CORE_AUDIO_DEVIN_APP_BUFFER_7	

enum AgsCoreAudioDevinSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_CORE_AUDIO_DEVIN_PASS_THROUGH	do not sync
AGS_CORE_AUDIO_DEVIN_INITIAL_CALLBACK	initial call-back
AGS_CORE_AUDIO_DEVIN_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_CORE_AUDIO_DEVIN_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_CORE_AUDIO_DEVIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_CORE_AUDIO_DEVIN_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_CORE_AUDIO_DEVIN_ERROR

```
#define AGS_CORE_AUDIO_DEVIN_ERROR (ags_core_audio_devin_error_quark())
```

enum AgsCoreAudioDevinError

Members

AGS_CORE_AUDIO_DEVIN_ERROR_LOCKED_SOUNDCARD | |

AGS_TYPE_CORE_AUDIO_DEVIN

```
#define AGS_TYPE_CORE_AUDIO_DEVIN                (ags_core_audio_devin_get_type())
```

struct AgsCoreAudioDevin

```
struct AgsCoreAudioDevin;
```

struct AgsCoreAudioDevinClass

```
struct AgsCoreAudioDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsCoreAudioDevin

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsCoreAudioDevin

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.0.0

The "core-audio-client" property

```
"core-audio-client"          AgsCoreAudioClient~*
```

The assigned AgsCoreAudioClient

Owner: AgsCoreAudioDevin

Flags: Read / Write

Since: 3.0.0

The "core-audio-port" property

```
"core-audio-port"            gpointer
```

The assigned AgsCoreAudioPort

Owner: AgsCoreAudioDevin

Flags: Read / Write

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"               double
```

tact

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                     char~*
```

The core audio soundcard indentifier

Owner: AgsCoreAudioDevin

Flags: Read / Write

Default value: "ags-core-audio-devin-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsCoreAudioDevin

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.0.0

8.30 AgsCoreAudioDevout

AgsCoreAudioDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CORE_AUDIO_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_core_audio_devout_error_quark ()
gboolean	ags_core_audio_devout_test_flags ()
void	ags_core_audio_devout_set_flags ()
void	ags_core_audio_devout_unset_flags ()
void	ags_core_audio_devout_switch_buffer_flag ()
void	ags_core_audio_devout_adjust_delay_and_attack ()
void	ags_core_audio_devout_realloc_buffer ()
AgsCoreAudioDevout *	ags_core_audio_devout_new ()
#define	AGS_CORE_AUDIO_DEVOUT()
#define	AGS_CORE_AUDIO_DEVOUT_CLASS()
#define	AGS_CORE_AUDIO_DEVOUT_GET_CLASS()
#define	AGS_IS_CORE_AUDIO_DEVOUT()
#define	AGS_IS_CORE_AUDIO_DEVOUT_CLASS()
GType	ags_core_audio_devout_flags_get_type ()
GType	ags_core_audio_devout_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
AgsCoreAudioClient *	core-audio-client	Read / Write
gpointer	core-audio-port	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_CORE_AUDIO_DEVOUT_DEFAULT_APP_BUFFER_SIZE
enum	AgsCoreAudioDevoutFlags
enum	AgsCore_AudioDevoutAppBufferMode
enum	AgsCoreAudioDevoutSyncFlags
#define	AGS_CORE_AUDIO_DEVOUT_ERROR
enum	AgsCoreAudioDevoutError
#define	AGS_TYPE_CORE_AUDIO_DEVOUT

struct	AgsCoreAudioDevout
struct	AgsCoreAudioDevoutClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsCoreAudioDevoutFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsCoreAudioDevout
```

Implemented Interfaces

AgsCoreAudioDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/core-audio/ags_core_audio_devout.h>
```

Description

AgsCoreAudioDevout represents a soundcard and supports output.

Functions

AGS_CORE_AUDIO_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_CORE_AUDIO_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsCoreAudioDevout *) obj)->obj_mutex ↩
    ))
```

ags_core_audio_devout_error_quark ()

```
GQuark
ags_core_audio_devout_error_quark ();
```

ags_core_audio_devout_test_flags ()

```
gboolean
ags_core_audio_devout_test_flags (AgsCoreAudioDevout *core_audio_devout,
                                  guint flags);
```

Test *flags* to be set on *core_audio_devout* .

Parameters

core_audio_devout	the AgsCoreAudioDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_core_audio_devout_set_flags ()

```
void
ags_core_audio_devout_set_flags (AgsCoreAudioDevout *core_audio_devout,
                                 guint flags);
```

Enable a feature of *core_audio_devout* .

Parameters

core_audio_devout	the AgsCoreAudioDevout	
flags	see AgsCoreAudioDevoutFlags	

Since: 3.0.0

ags_core_audio_devout_unset_flags ()

```
void
ags_core_audio_devout_unset_flags (AgsCoreAudioDevout *core_audio_devout,
                                   guint flags);
```

Disable a feature of *core_audio_devout* .

Parameters

core_audio_devout	the AgsCoreAudioDevout	
flags	see AgsCoreAudioDevoutFlags	

Since: 3.0.0

ags_core_audio_devout_switch_buffer_flag ()

```
void
ags_core_audio_devout_switch_buffer_flag
                                (AgsCoreAudioDevout *core_audio_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

core_audio_devout	an AgsCoreAudioDevout	

Since: 3.0.0

ags_core_audio_devout_adjust_delay_and_attack ()

```
void
ags_core_audio_devout_adjust_delay_and_attack
                                (AgsCoreAudioDevout *core_audio_devout);
```

Calculate delay and attack and reset it.

Parameters

core_audio_devout	the AgsCoreAudioDevout

Since: 3.0.0

ags_core_audio_devout_realloc_buffer ()

```
void
ags_core_audio_devout_realloc_buffer (AgsCoreAudioDevout *core_audio_devout);
```

Reallocate the internal audio buffer.

Parameters

core_audio_devout	the AgsCoreAudioDevout

Since: 3.0.0

ags_core_audio_devout_new ()

```
AgsCoreAudioDevout~*
ags_core_audio_devout_new ();
```

Creates a new instance of AgsCoreAudioDevout.

Returns

a new AgsCoreAudioDevout

Since: 3.0.0

AGS_CORE_AUDIO_DEVOUT()

```
#define AGS_CORE_AUDIO_DEVOUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CORE_AUDIO_DEVOUT, AgsCoreAudioDevout))
```

AGS_CORE_AUDIO_DEVOUT_CLASS()

```
#define AGS_CORE_AUDIO_DEVOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CORE_AUDIO_DEVOUT, AgsCoreAudioDevout))
```

AGS_CORE_AUDIO_DEVOUT_GET_CLASS()

```
#define AGS_CORE_AUDIO_DEVOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CORE_AUDIO_DEVOUT, AgsCoreAudioDevoutClass))
```

AGS_IS_CORE_AUDIO_DEVOUT()

```
#define AGS_IS_CORE_AUDIO_DEVOUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CORE_AUDIO_DEVOUT))
```

AGS_IS_CORE_AUDIO_DEVOUT_CLASS()

```
#define AGS_IS_CORE_AUDIO_DEVOUT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CORE_AUDIO_DEVOUT))
```

ags_core_audio_devout_flags_get_type ()

```
GType
ags_core_audio_devout_flags_get_type ();
```

ags_core_audio_devout_get_type ()

```
GType
ags_core_audio_devout_get_type ();
```

Types and Values

AGS_CORE_AUDIO_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_CORE_AUDIO_DEVOUT_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsCoreAudioDevoutFlags

Enum values to control the behavior or indicate internal state of AgsCoreAudioDevout by enable/disable as flags.

Members

AGS_CORE_AUDIO_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_CORE_AUDIO_DEVOUT_START_PLAY	playback start-ing

AGS_CORE_AUDIO_DEVOUT_PLAY	do play-back
AGS_CORE_AUDIO_DEVOUT_SHUTDOWN	stop play-back
AGS_CORE_AUDIO_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_CORE_AUDIO_DEVOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsCore_AudioDevoutAppBufferMode

Members

AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_0		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_1		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_2		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_3		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_4		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_5		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_6		
AGS_CORE_AUDIO_DEVOUT_APP_BUFFER_7		

enum AgsCoreAudioDevoutSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_CORE_AUDIO_DEVOUT_PASS_THROUGH	do not sync
AGS_CORE_AUDIO_DEVOUT_INITIAL_CALLBACK	initial call-back
AGS_CORE_AUDIO_DEVOUT_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock

AGS_CORE_AUDIO_DEVOUT_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_CORE_AUDIO_DEVOUT_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_CORE_AUDIO_DEVOUT_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock

AGS_CORE_AUDIO_DEVOUT_ERROR

```
#define AGS_CORE_AUDIO_DEVOUT_ERROR (ags_core_audio_devout_error_quark())
```

enum AgsCoreAudioDevoutError

Members

AGS_CORE_AUDIO_DEVOUT_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_CORE_AUDIO_DEVOUT

```
#define AGS_TYPE_CORE_AUDIO_DEVOUT                (ags_core_audio_devout_get_type())
```

struct AgsCoreAudioDevout

```
struct AgsCoreAudioDevout;
```

struct AgsCoreAudioDevoutClass

```
struct AgsCoreAudioDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
"attack"                    gpointer
```

Attack of the buffer

Owner: AgsCoreAudioDevout

Flags: Read

Since: 3.0.0

The "bpm" property

```
"bpm"                       double
```

Beats per minute

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
"buffer"                    gpointer
```

The buffer

Owner: AgsCoreAudioDevout

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [16,16384]

Default value: 1024

Since: 3.0.0

The "core-audio-client" property

```
"core-audio-client"        AgsCoreAudioClient~*
```

The assigned AgsCoreAudioClient

Owner: AgsCoreAudioDevout

Flags: Read / Write

Since: 3.0.0

The "core-audio-port" property

```
"core-audio-port"          gpointer
```

The assigned AgsCoreAudioPort

Owner: AgsCoreAudioDevout

Flags: Read / Write

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"             double
```

tact

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                   char~*
```

The core audio soundcard indentifier

Owner: AgsCoreAudioDevout

Flags: Read / Write

Default value: "ags-core-audio-devout-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"             guint
```

The dsp channel count

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsCoreAudioDevout

Flags: Read / Write

Allowed values: [8000,5644800]

Default value: 48000

Since: 3.0.0

8.31 AgsCoreAudioMidiin

AgsCoreAudioMidiin — Input from sequencer

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CORE_AUDIO_MIDIIN_GET_OBJ_MUTEX()
GQuark	ags_core_audio_midiin_error_quark ()
gboolean	ags_core_audio_midiin_test_flags ()
void	ags_core_audio_midiin_set_flags ()
void	ags_core_audio_midiin_unset_flags ()
void	ags_core_audio_midiin_switch_buffer_flag ()
AgsCoreAudioMidiin *	ags_core_audio_midiin_new ()
#define	AGS_CORE_AUDIO_MIDIIN()
#define	AGS_CORE_AUDIO_MIDIIN_CLASS()
#define	AGS_CORE_AUDIO_MIDIIN_GET_CLASS()
#define	AGS_IS_CORE_AUDIO_MIDIIN()
#define	AGS_IS_CORE_AUDIO_MIDIIN_CLASS()
GType	ags_core_audio_midiin_flags_get_type ()
GType	ags_core_audio_midiin_get_type ()

Properties

gpointer	attack	Read
double	bpm	Read / Write
gpointer	buffer	Read
AgsCoreAudioClient *	core-audio-client	Read / Write
AgsCoreAudioPort *	core-audio-port	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write

Types and Values

#define	AGS_CORE_AUDIO_MIDIIN_DEFAULT_BUFFER_SIZE
enum	AgsCoreAudioMidiinFlags
enum	AgsCoreAudioMidiinAppBufferMode
enum	AgsCoreAudioMidiinSyncFlags
#define	AGS_CORE_AUDIO_MIDIIN_ERROR
enum	AgsCoreAudioMidiinError
#define	AGS_TYPE_CORE_AUDIO_MIDIIN
struct	AgsCoreAudioMidiin
struct	AgsCoreAudioMidiinClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsCoreAudioMidiinFlags
GObject
&#x2570;&#x2500;&#x2500; AgsCoreAudioMidiin
```

Implemented Interfaces

AgsCoreAudioMidiin implements AgsConnectable and AgsSequencer.

Includes

```
#include <ags/audio/core-audio/ags_core_audio_midiin.h>
```

Description

AgsCoreAudioMidiin represents a sequencer and supports midi input.

Functions

AGS_CORE_AUDIO_MIDIIN_GET_OBJ_MUTEX()

```
#define AGS_CORE_AUDIO_MIDIIN_GET_OBJ_MUTEX(obj) (&(((AgsCoreAudioMidiin *) obj)->obj_mutex ↩
    ))
```

ags_core_audio_midiin_error_quark ()

```
GQuark
ags_core_audio_midiin_error_quark ();
```

ags_core_audio_midiin_test_flags ()

```
gboolean
ags_core_audio_midiin_test_flags (AgsCoreAudioMidiin *core_audio_midiin,
                                  guint flags);
```

Test *flags* to be set on *core_audio_midiin* .

Parameters

core_audio_midiin	the AgsCoreAudioMidiin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_core_audio_midiin_set_flags ()

```
void
ags_core_audio_midiin_set_flags (AgsCoreAudioMidiin *core_audio_midiin,
                                 guint flags);
```

Enable a feature of *core_audio_midiin* .

Parameters

core_audio_midiin	the AgsCoreAudioMidiin	
flags	see AgsCoreAudioMidiinFlags	

Since: 3.0.0

ags_core_audio_midiin_unset_flags ()

```
void
ags_core_audio_midiin_unset_flags (AgsCoreAudioMidiin *core_audio_midiin,
                                   guint flags);
```

Disable a feature of *core_audio_midiin* .

Parameters

core_audio_midiin	the AgsCoreAudioMidiin	
flags	see AgsCoreAudioMidiinFlags	

Since: 3.0.0

ags_core_audio_midiin_switch_buffer_flag ()

```
void
ags_core_audio_midiin_switch_buffer_flag
                                (AgsCoreAudioMidiin *core_audio_midiin);
```

The buffer flag indicates the currently recorded buffer.

Parameters

core_audio_midiin	the AgsCoreAudioMidiin	

Since: 3.0.0

ags_core_audio_midiin_new ()

```
AgsCoreAudioMidiin~*
ags_core_audio_midiin_new ();
```

Creates a new instance of AgsCoreAudioMidiin.

Returns

a new AgsCoreAudioMidiin

Since: 3.0.0

AGS_CORE_AUDIO_MIDIIN()

```
#define AGS_CORE_AUDIO_MIDIIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CORE_AUDIO_MIDIIN, AgsCoreAudioMidiin))
```

AGS_CORE_AUDIO_MIDIIN_CLASS()

```
#define AGS_CORE_AUDIO_MIDIIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CORE_AUDIO_MIDIIN, AgsCoreAudioMidiin))
```

AGS_CORE_AUDIO_MIDIIN_GET_CLASS()

```
#define AGS_CORE_AUDIO_MIDIIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CORE_AUDIO_MIDIIN, AgsCoreAudioMidiinClass))
```

AGS_IS_CORE_AUDIO_MIDIIN()

```
#define AGS_IS_CORE_AUDIO_MIDIIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CORE_AUDIO_MIDIIN))
```

AGS_IS_CORE_AUDIO_MIDIIN_CLASS()

```
#define AGS_IS_CORE_AUDIO_MIDIIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CORE_AUDIO_MIDIIN))
```

ags_core_audio_midiin_flags_get_type ()

```
GType
ags_core_audio_midiin_flags_get_type ();
```

ags_core_audio_midiin_get_type ()

```
GType
ags_core_audio_midiin_get_type ();
```

Types and Values

AGS_CORE_AUDIO_MIDIIN_DEFAULT_BUFFER_SIZE

```
#define AGS_CORE_AUDIO_MIDIIN_DEFAULT_BUFFER_SIZE (256)
```

enum AgsCoreAudioMidiinFlags

Enum values to control the behavior or indicate internal state of AgsCoreAudioMidiin by enable/disable as flags.

Members

AGS_CORE_AUDIO_MIDIIN_INITIALIZED	recording is initialized
AGS_CORE_AUDIO_MIDIIN_START_RECORD	just started recording

AGS_CORE_AUDIO_MIDIIN_RECORD	is recording
AGS_CORE_AUDIO_MIDIIN_SHUTDOWN	stop recording
AGS_CORE_AUDIO_MIDIIN_NONBLOCKING	do non-blocking calls
AGS_CORE_AUDIO_MIDIIN_ATTACK_FIRST	use first attack, instead of second one

enum AgsCoreAudioMidiinAppBufferMode

Enum values to indicate internal state of AgsCoreAudioMidiin application buffer by setting mode.

Members

AGS_CORE_AUDIO_MIDIIN_APP_BUFFER_0	ring-buffer 0
AGS_CORE_AUDIO_MIDIIN_APP_BUFFER_1	ring-buffer 1
AGS_CORE_AUDIO_MIDIIN_APP_BUFFER_2	ring-buffer 2
AGS_CORE_AUDIO_MIDIIN_APP_BUFFER_3	ring-buffer 3

enum AgsCoreAudioMidiinSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_CORE_AUDIO_MIDIIN_PASS_THROUGH	do not sync
AGS_CORE_AUDIO_MIDIIN_INITIAL_CALLBACK	initial callback

AGS_CORE_AUDIO_MIDIIN_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_CORE_AUDIO_MIDIIN_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_CORE_AUDIO_MIDIIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock
AGS_CORE_AUDIO_MIDIIN_CALLBACK_FINISH_DONE	sync done, client con-di-tional lock
AGS_CORE_AUDIO_MIDIIN_DO_SYNC	do syn-chro-nize

AGS_CORE_AUDIO_MIDIIN_ERROR

```
#define AGS_CORE_AUDIO_MIDIIN_ERROR (ags_core_audio_midiin_error_quark())
```

enum AgsCoreAudioMidiinError

Members

AGS_CORE_AUDIO_MIDIIN_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_CORE_AUDIO_MIDIIN

```
#define AGS_TYPE_CORE_AUDIO_MIDIIN                (ags_core_audio_midiin_get_type())
```

struct AgsCoreAudioMidiin

```
struct AgsCoreAudioMidiin;
```

struct AgsCoreAudioMidiinClass

```
struct AgsCoreAudioMidiinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsCoreAudioMidiin

Flags: Read

Since: 3.0.0

The "bpm" property

```
  "bpm"                       double
```

Beats per minute

Owner: AgsCoreAudioMidiin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
  "buffer"                    gpointer
```

The buffer

Owner: AgsCoreAudioMidiin

Flags: Read

Since: 3.0.0

The "core-audio-client" property

```
  "core-audio-client"         AgsCoreAudioClient~*
```

The assigned AgsCoreAudioClient

Owner: AgsCoreAudioMidiin

Flags: Read / Write

Since: 3.0.0

The "core-audio-port" property

```
"core-audio-port"          AgsCoreAudioPort~*
```

The assigned AgsCoreAudioPort

Owner: AgsCoreAudioMidiin

Flags: Read / Write

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"             double
```

tact

Owner: AgsCoreAudioMidiin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                   char~*
```

The core audio sequencer indentifier

Owner: AgsCoreAudioMidiin

Flags: Read / Write

Default value: "hw:0"

Since: 3.0.0

8.32 AgsAudioUnitServer

AgsAudioUnitServer — audio unit instance

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_UNIT_SERVER_GET_OBJ_MUTEX()
gboolean	ags_audio_unit_server_test_flags ()
void	ags_audio_unit_server_set_flags ()
void	ags_audio_unit_server_unset_flags ()

GList *	ags_audio_unit_server_find_url ()
GObject *	ags_audio_unit_server_find_client ()
GObject *	ags_audio_unit_server_find_port ()
void	ags_audio_unit_server_add_client ()
void	ags_audio_unit_server_remove_client ()
void	ags_audio_unit_server_connect_client ()
void	ags_audio_unit_server_start_poll ()
AgsAudioUnitServer *	ags_audio_unit_server_new ()
#define	AGS_AUDIO_UNIT_SERVER()
#define	AGS_AUDIO_UNIT_SERVER_CLASS()
#define	AGS_AUDIO_UNIT_SERVER_GET_CLASS()
#define	AGS_IS_AUDIO_UNIT_SERVER()
#define	AGS_IS_AUDIO_UNIT_SERVER_CLASS()
GType	ags_audio_unit_server_get_type ()

Properties

gpointer	audio-unit-client	Read / Write
AgsAudioUnitClient *	default-audio-unit-client	Read / Write
GObject *	default-soundcard	Read / Write
AgsAudioUnitClient *	input-audio-unit-client	Read / Write
char *	url	Read / Write

Types and Values

#define	AGS_TYPE_AUDIO_UNIT_SERVER
struct	AgsAudioUnitServer
struct	AgsAudioUnitServerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsAudioUnitServer
```

Implemented Interfaces

AgsAudioUnitServer implements AgsConnectable and AgsSoundServer.

Includes

```
#include <ags/audio/audio-unit/ags_audio_unit_server.h>
```

Description

The AgsAudioUnitServer is an object to represent a running audio unit instance.

Functions

AGS_AUDIO_UNIT_SERVER_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_UNIT_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsAudioUnitServer *) obj)->obj_mutex ↩
    ))
```

ags_audio_unit_server_test_flags ()

```
gboolean
ags_audio_unit_server_test_flags (AgsAudioUnitServer *audio_unit_server,
                                  guint flags);
```

Test *flags* to be set on *audio_unit_server* .

Parameters

audio_unit_server	the AgsAudioUnitServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_unit_server_set_flags ()

```
void
ags_audio_unit_server_set_flags (AgsAudioUnitServer *audio_unit_server,
                                 guint flags);
```

Enable a feature of *audio_unit_server* .

Parameters

audio_unit_server	the AgsAudioUnitServer	
flags	see AgsAudioUnitServerFlags	

Since: 3.0.0

ags_audio_unit_server_unset_flags ()

```
void
ags_audio_unit_server_unset_flags (AgsAudioUnitServer *audio_unit_server,
                                   guint flags);
```

Disable a feature of *audio_unit_server* .

Parameters

audio_unit_server	the AgsAudioUnitServer	
flags	see AgsAudioUnitServerFlags	

Since: 3.0.0

ags_audio_unit_server_find_url ()

```
GList~*
ags_audio_unit_server_find_url (GList *audio_unit_server,
                                gchar *url);
```

Find AgsAudioUnitServer by url.

Parameters

audio_unit_server	the GList containing AgsAudioUnitServer.	*[element-type AgsAudio.AudioUnitServer][transfer none]*
url	the url to find	

Returns

the next matching GList containing a AgsAudioUnitServer matching `url` or NULL.

[element-type AgsAudio.AudioUnitServer][transfer none]

Since: 3.0.0

ags_audio_unit_server_find_client ()

```
GObject~*
ags_audio_unit_server_find_client (AgsAudioUnitServer *audio_unit_server,
                                   gchar *client_uuid);
```

Find AgsAudioUnitClient by uuid.

Parameters

audio_unit_server	the AgsAudioUnitServer	
client_uuid	the uuid to find	

Returns

the AgsAudioUnitClient found or NULL.

[transfer none]

Since: 3.0.0

ags_audio_unit_server_find_port ()

```
GObject~*
ags_audio_unit_server_find_port (AgsAudioUnitServer *audio_unit_server,
                                 gchar *port_uuid);
```

Find AgsAudioUnitPort by uuid.

Parameters

audio_unit_server	the AgsAudioUnitServer	
port_uuid	the uuid to find	

Returns

the AgsAudioUnitPort found or NULL.

[transfer none]

Since: 3.0.0

ags_audio_unit_server_add_client ()

```
void
ags_audio_unit_server_add_client (AgsAudioUnitServer *audio_unit_server,
                                  GObject *audio_unit_client);
```

Add *audio_unit_client* to *audio_unit_server*

Parameters

audio_unit_server	the AgsAudioUnitServer	
audio_unit_client	the AgsAudioUnitClient to add	

Since: 3.0.0

ags_audio_unit_server_remove_client ()

```
void
ags_audio_unit_server_remove_client (AgsAudioUnitServer *audio_unit_server,
                                     GObject *audio_unit_client);
```

Remove *audio_unit_client* to *audio_unit_server*

Parameters

audio_unit_server	the AgsAudioUnitServer	
audio_unit_client	the AgsAudioUnitClient to remove	

Since: 3.0.0

ags_audio_unit_server_connect_client ()

```
void
ags_audio_unit_server_connect_client (AgsAudioUnitServer *audio_unit_server);
```

Connect all clients.

Parameters

audio_unit_server	the AgsAudioUnitServer

Since: 3.0.0

ags_audio_unit_server_start_poll ()

```
void
ags_audio_unit_server_start_poll (AgsAudioUnitServer *audio_unit_server);
```

ags_audio_unit_server_new ()

```
AgsAudioUnitServer~*
ags_audio_unit_server_new (gchar *url);
```

Create a new instance of AgsAudioUnitServer.

Parameters

url	the URL as string

Returns

the new AgsAudioUnitServer

Since: 3.0.0

AGS_AUDIO_UNIT_SERVER()

```
#define AGS_AUDIO_UNIT_SERVER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_SERVER, AgsAudioUnitServer))
```

AGS_AUDIO_UNIT_SERVER_CLASS()

```
#define AGS_AUDIO_UNIT_SERVER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_UNIT_SERVER, AgsAudioUnitServer))
```

AGS_AUDIO_UNIT_SERVER_GET_CLASS()

```
#define AGS_AUDIO_UNIT_SERVER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_UNIT_SERVER, AgsAudioUnitServerClass))
```

AGS_IS_AUDIO_UNIT_SERVER()

```
#define AGS_IS_AUDIO_UNIT_SERVER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_SERVER))
```

AGS_IS_AUDIO_UNIT_SERVER_CLASS()

```
#define AGS_IS_AUDIO_UNIT_SERVER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_UNIT_SERVER))
```

ags_audio_unit_server_get_type ()

```
GType
ags_audio_unit_server_get_type ();
```

Types and Values

AGS_TYPE_AUDIO_UNIT_SERVER

```
#define AGS_TYPE_AUDIO_UNIT_SERVER                 (ags_audio_unit_server_get_type())
```

struct AgsAudioUnitServer

```
struct AgsAudioUnitServer;
```

struct AgsAudioUnitServerClass

```
struct AgsAudioUnitServerClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-unit-client" property

```
  "audio-unit-client"        gpointer
```

The audio unit client list.

[transfer full]

Owner: AgsAudioUnitServer

Flags: Read / Write

Since: 3.0.0

The "default-audio-unit-client" property

```
"default-audio-unit-client" AgsAudioUnitClient~*
```

The default audio unit client.

Owner: AgsAudioUnitServer

Flags: Read / Write

Since: 3.0.0

The "default-soundcard" property

```
"default-soundcard"         GObject~*
```

The default soundcard.

Owner: AgsAudioUnitServer

Flags: Read / Write

Since: 3.0.0

The "input-audio-unit-client" property

```
"input-audio-unit-client"  AgsAudioUnitClient~*
```

The input audio unit client.

Owner: AgsAudioUnitServer

Flags: Read / Write

Since: 3.0.0

The "url" property

```
"url"                       char~*
```

The assigned URL.

Owner: AgsAudioUnitServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

8.33 AgsAudioUnitClient

AgsAudioUnitClient — core audio connection

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_UNIT_CLIENT_GET_OBJ_MUTEX()
gboolean	ags_audio_unit_client_test_flags ()
void	ags_audio_unit_client_set_flags ()
void	ags_audio_unit_client_unset_flags ()
GList *	ags_audio_unit_client_find_uuid ()
GList *	ags_audio_unit_client_find ()
void	ags_audio_unit_client_open ()
void	ags_audio_unit_client_add_device ()
void	ags_audio_unit_client_remove_device ()
void	ags_audio_unit_client_add_port ()
void	ags_audio_unit_client_remove_port ()
void	ags_audio_unit_client_activate ()
void	ags_audio_unit_client_deactivate ()
AgsAudioUnitClient *	ags_audio_unit_client_new ()
#define	AGS_AUDIO_UNIT_CLIENT()
#define	AGS_AUDIO_UNIT_CLIENT_CLASS()
#define	AGS_AUDIO_UNIT_CLIENT_GET_CLASS()
#define	AGS_IS_AUDIO_UNIT_CLIENT()
#define	AGS_IS_AUDIO_UNIT_CLIENT_CLASS()
GType	ags_audio_unit_client_flags_get_type ()
GType	ags_audio_unit_client_get_type ()

Properties

AgsAudioUnitServer *	audio-unit-server	Read / Write
char *	client-name	Read / Write
gpointer	device	Read / Write
gpointer	port	Read / Write

Types and Values

enum	AgsAudioUnitClientFlags
#define	AGS_TYPE_AUDIO_UNIT_CLIENT
struct	AgsAudioUnitClient
struct	AgsAudioUnitClientClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsAudioUnitClientFlags
GObject
&#x2570;&#x2500;&#x2500; AgsAudioUnitClient
```

Implemented Interfaces

AgsAudioUnitClient implements AgsConnectable.

Includes

```
#include <ags/audio/audio-unit/ags_audio_unit_client.h>
```

Description

The AgsAudioUnitClient communicates with a core audio instance.

Functions

AGS_AUDIO_UNIT_CLIENT_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_UNIT_CLIENT_GET_OBJ_MUTEX(obj) (&(((AgsAudioUnitClient *) obj)->obj_mutex ↩
    ))
```

ags_audio_unit_client_test_flags ()

```
gboolean
ags_audio_unit_client_test_flags (AgsAudioUnitClient *audio_unit_client,
                                  guint flags);
```

Test *flags* to be set on *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_unit_client_set_flags ()

```
void
ags_audio_unit_client_set_flags (AgsAudioUnitClient *audio_unit_client,
                                 guint flags);
```

Enable a feature of *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
flags	see AgsAudioUnitClientFlags	

Since: 3.0.0

ags_audio_unit_client_unset_flags ()

```
void
ags_audio_unit_client_unset_flags (AgsAudioUnitClient *audio_unit_client,
                                   guint flags);
```

Disable a feature of *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
flags	see AgsAudioUnitClientFlags	

Since: 3.0.0

ags_audio_unit_client_find_uuid ()

```
GList~*
ags_audio_unit_client_find_uuid (GList *audio_unit_client,
                                 gchar *client_uuid);
```

Finds next match of `client_uuid` in `audio_unit_client` .

Parameters

audio_unit_client	the GList containing AgsAudioUnitClient.	*[element-type AgsAudio.AudioUnitClient][transfer none]*
client_uuid	the client uuid to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.AudioUnitClient][transfer none]

Since: 3.0.0

ags_audio_unit_client_find ()

```
GList~*
ags_audio_unit_client_find (GList *audio_unit_client,
                            gchar *client_name);
```

Finds next match of `client_name` in `audio_unit_client` .

Parameters

audio_unit_client	the GList containing AgsAudioUnitClient.	*[element-type AgsAudio.AudioUnitClient][transfer none]*
client_name	the client name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.AudioUnitClient][transfer none]

Since: 3.0.0

ags_audio_unit_client_open ()

```
void
ags_audio_unit_client_open (AgsAudioUnitClient *audio_unit_client,
                            gchar *client_name);
```

Open the core audio client's connection and read uuid.

Parameters

audio_unit_client	the AgsAudioUnitClient	
client_name	the client's name	

Since: 3.0.0

ags_audio_unit_client_add_device ()

```
void
ags_audio_unit_client_add_device (AgsAudioUnitClient *audio_unit_client,
                                  GObject *audio_unit_device);
```

Add *audio_unit_device* to *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
audio_unit_device	an AgsAudioUnitDevout or AgsAudioUnitDevin	

Since: 3.0.0

ags_audio_unit_client_remove_device ()

```
void
ags_audio_unit_client_remove_device (AgsAudioUnitClient *audio_unit_client,
                                     GObject *audio_unit_device);
```

Remove *audio_unit_device* from *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
audio_unit_device	an AgsAudioUnitDevout or AgsAudioUnitDevin	

Since: 3.0.0

ags_audio_unit_client_add_port ()

```
void
ags_audio_unit_client_add_port (AgsAudioUnitClient *audio_unit_client,
                                GObject *audio_unit_port);
```

Add *audio_unit_port* to *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
audio_unit_port	an AgsAudioUnitPort	

Since: 3.0.0

ags_audio_unit_client_remove_port ()

```
void
ags_audio_unit_client_remove_port (AgsAudioUnitClient *audio_unit_client,
                                   GObject *audio_unit_port);
```

Remove *audio_unit_port* from *audio_unit_client* .

Parameters

audio_unit_client	the AgsAudioUnitClient	
audio_unit_port	an AgsAudioUnitPort	

Since: 3.0.0

ags_audio_unit_client_activate ()

```
void
ags_audio_unit_client_activate (AgsAudioUnitClient *audio_unit_client);
```

Activate client.

Parameters

audio_unit_client	the AgsAudioUnitClient	

Since: 3.0.0

ags_audio_unit_client_deactivate ()

```
void
ags_audio_unit_client_deactivate (AgsAudioUnitClient *audio_unit_client);
```

Deactivate client.

Parameters

audio_unit_client	the AgsAudioUnitClient	

Since: 3.0.0

ags_audio_unit_client_new ()

```
AgsAudioUnitClient~*
ags_audio_unit_client_new (GObject *audio_unit_server);
```

Create a new instance of AgsAudioUnitClient.

Parameters

audio_unit_server	the assigned AgsAudioUnitServer

Returns

the new AgsAudioUnitClient

Since: 3.0.0

AGS_AUDIO_UNIT_CLIENT()

```
#define AGS_AUDIO_UNIT_CLIENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_CLIENT, AgsAudioUnitClient))
```

AGS_AUDIO_UNIT_CLIENT_CLASS()

```
#define AGS_AUDIO_UNIT_CLIENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_UNIT_CLIENT, AgsAudioUnitClient))
```

AGS_AUDIO_UNIT_CLIENT_GET_CLASS()

```
#define AGS_AUDIO_UNIT_CLIENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_UNIT_CLIENT, AgsAudioUnitClientClass))
```

AGS_IS_AUDIO_UNIT_CLIENT()

```
#define AGS_IS_AUDIO_UNIT_CLIENT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_CLIENT))
```

AGS_IS_AUDIO_UNIT_CLIENT_CLASS()

```
#define AGS_IS_AUDIO_UNIT_CLIENT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_UNIT_CLIENT))
```

ags_audio_unit_client_flags_get_type ()

```
GType
ags_audio_unit_client_flags_get_type ();
```

ags_audio_unit_client_get_type ()

```
GType
ags_audio_unit_client_get_type ();
```

Types and Values

enum AgsAudioUnitClientFlags

Enum values to control the behavior or indicate internal state of AgsAudioUnitClient by enable/disable as flags.

Members

AGS_AUDIO_UNIT_CLIENT_ACTIVATED	the client was activated
AGS_AUDIO_UNIT_CLIENT_READY	the client is ready

AGS_TYPE_AUDIO_UNIT_CLIENT

```
#define AGS_TYPE_AUDIO_UNIT_CLIENT                (ags_audio_unit_client_get_type())
```

struct AgsAudioUnitClient

```
struct AgsAudioUnitClient;
```

struct AgsAudioUnitClientClass

```
struct AgsAudioUnitClientClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-unit-server" property

```
  "audio-unit-server"         AgsAudioUnitServer~*
```

The assigned AgsAudioUnitServer.

Owner: AgsAudioUnitClient

Flags: Read / Write

Since: 3.0.0

The "client-name" property

```
"client-name"                char~*
```

The audio-unit client name.

Owner: AgsAudioUnitClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "device" property

```
"device"                     gpointer
```

The assigned devices.

[transfer full]

Owner: AgsAudioUnitClient

Flags: Read / Write

Since: 3.0.0

The "port" property

```
"port"                       gpointer
```

The assigned ports.

[transfer full]

Owner: AgsAudioUnitClient

Flags: Read / Write

Since: 3.0.0

8.34 AgsAudioUnitPort

AgsAudioUnitPort — core audio resource.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_UNIT_PORT_GET_OBJ_MUTEX()
gboolean	ags_audio_unit_port_test_flags ()
void	ags_audio_unit_port_set_flags ()
void	ags_audio_unit_port_unset_flags ()

GList *	ags_audio_unit_port_find ()
void	ags_audio_unit_port_register ()
void	ags_audio_unit_port_unregister ()
void	ags_audio_unit_port_set_format ()
void	ags_audio_unit_port_set_samplerate ()
void	ags_audio_unit_port_set_pcm_channels ()
void	ags_audio_unit_port_set_buffer_size ()
AgsAudioUnitPort *	ags_audio_unit_port_new ()
#define	AGS_AUDIO_UNIT_PORT()
#define	AGS_AUDIO_UNIT_PORT_CLASS()
#define	AGS_AUDIO_UNIT_PORT_GET_CLASS()
#define	AGS_IS_AUDIO_UNIT_PORT()
#define	AGS_IS_AUDIO_UNIT_PORT_CLASS()
GType	ags_audio_unit_port_flags_get_type ()
GType	ags_audio_unit_port_get_type ()

Properties

AgsAudioUnitClient *	audio-unit-client	Read / Write
GObject *	audio-unit-device	Read / Write
char *	port-name	Read / Write

Types and Values

#define	AGS_AUDIO_UNIT_PORT_DEFAULT_CACHE_BUFFER_SIZE
enum	AgsAudioUnitPortFlags
#define	AGS_TYPE_AUDIO_UNIT_PORT
struct	AgsAudioUnitPort
struct	AgsAudioUnitPortClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsAudioUnitPortFlags
GObject
&#x2570;&#x2500;&#x2500; AgsAudioUnitPort
```

Implemented Interfaces

AgsAudioUnitPort implements AgsConnectable.

Includes

```
#include <ags/audio/audio-unit/ags_audio_unit_port.h>
```

Description

The AgsAudioUnitPort represents either a core audio sequencer or soundcard to communicate with.

Functions

AGS_AUDIO_UNIT_PORT_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_UNIT_PORT_GET_OBJ_MUTEX(obj) (&(((AgsAudioUnitPort *) obj)->obj_mutex))
```

ags_audio_unit_port_test_flags ()

```
gboolean
ags_audio_unit_port_test_flags (AgsAudioUnitPort *audio_unit_port,
                                guint flags);
```

Test *flags* to be set on *audio_unit_port* .

Parameters

audio_unit_port	the AgsAudioUnitPort	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_unit_port_set_flags ()

```
void
ags_audio_unit_port_set_flags (AgsAudioUnitPort *audio_unit_port,
                               guint flags);
```

Enable a feature of *audio_unit_port* .

Parameters

audio_unit_port	the AgsAudioUnitPort	
flags	see AgsAudioUnitPortFlags	

Since: 3.0.0

ags_audio_unit_port_unset_flags ()

```
void
ags_audio_unit_port_unset_flags (AgsAudioUnitPort *audio_unit_port,
                                 guint flags);
```

Disable a feature of *audio_unit_port* .

Parameters

audio_unit_port	the AgsAudioUnitPort	
flags	see AgsAudioUnitPortFlags	

Since: 3.0.0

ags_audio_unit_port_find ()

```
GList~*
ags_audio_unit_port_find (GList *audio_unit_port,
                          gchar *port_name);
```

Finds next match of *port_name* in *audio_unit_port* .

Parameters

audio_unit_port	the GList containing AgsAudioUnitPort.	*[element-type AgsAudio.AudioUnitPort][transfer none]*
port_name	the port name to find	

Returns

the next matching GList or NULL.

[element-type AgsAudio.AudioUnitPort][transfer none]

Since: 3.0.0

ags_audio_unit_port_register ()

```
void
ags_audio_unit_port_register (AgsAudioUnitPort *audio_unit_port,
                              gchar *port_name,
                              gboolean is_audio,
                              gboolean is_midi,
                              gboolean is_output);
```

ags_audio_unit_port_unregister ()

```
void
ags_audio_unit_port_unregister (AgsAudioUnitPort *audio_unit_port);
```

ags_audio_unit_port_set_format ()

```
void
ags_audio_unit_port_set_format (AgsAudioUnitPort *audio_unit_port,
                                guint format);
```

ags_audio_unit_port_set_samplerate ()

```
void
ags_audio_unit_port_set_samplerate (AgsAudioUnitPort *audio_unit_port,
                                    guint samplerate);
```

ags_audio_unit_port_set_pcm_channels ()

```
void
ags_audio_unit_port_set_pcm_channels (AgsAudioUnitPort *audio_unit_port,
                                      guint pcm_channels);
```

ags_audio_unit_port_set_buffer_size ()

```
void
ags_audio_unit_port_set_buffer_size (AgsAudioUnitPort *audio_unit_port,
                                     guint buffer_size);
```

ags_audio_unit_port_new ()

```
AgsAudioUnitPort~*
ags_audio_unit_port_new (GObject *audio_unit_client);
```

Create a new instance of AgsAudioUnitPort.

Parameters

audio_unit_client	the AgsAudioUnitClient assigned to

Returns

the new AgsAudioUnitPort

Since: 3.0.0

AGS_AUDIO_UNIT_PORT()

```
#define AGS_AUDIO_UNIT_PORT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_PORT, AgsAudioUnitPort))
```

AGS_AUDIO_UNIT_PORT_CLASS()

```
#define AGS_AUDIO_UNIT_PORT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_UNIT_PORT, AgsAudioUnitPort))
```

AGS_AUDIO_UNIT_PORT_GET_CLASS()

```
#define AGS_AUDIO_UNIT_PORT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_UNIT_PORT, AgsAudioUnitPortClass))
```

AGS_IS_AUDIO_UNIT_PORT()

```
#define AGS_IS_AUDIO_UNIT_PORT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_PORT))
```

AGS_IS_AUDIO_UNIT_PORT_CLASS()

```
#define AGS_IS_AUDIO_UNIT_PORT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_UNIT_PORT))
```

ags_audio_unit_port_flags_get_type ()

```
GType
ags_audio_unit_port_flags_get_type ();
```

ags_audio_unit_port_get_type ()

```
GType
ags_audio_unit_port_get_type ();
```

Types and Values

AGS_AUDIO_UNIT_PORT_DEFAULT_CACHE_BUFFER_SIZE

```
#define AGS_AUDIO_UNIT_PORT_DEFAULT_CACHE_BUFFER_SIZE (4096)
```

enum AgsAudioUnitPortFlags

Enum values to control the behavior or indicate internal state of AgsAudioUnitPort by enable/disable as flags.

Members

AGS_AUDIO_UNIT_PORT_REGISTERED	the port was registered
AGS_AUDIO_UNIT_PORT_IS_AUDIO	the port provides audio data

AGS_AUDIO_UNIT_PORT_IS_MIDI	the port provides midi data
AGS_AUDIO_UNIT_PORT_IS_OUTPUT	the port does output
AGS_AUDIO_UNIT_PORT_IS_INPUT	the port does input

AGS_TYPE_AUDIO_UNIT_PORT

```
#define AGS_TYPE_AUDIO_UNIT_PORT                (ags_audio_unit_port_get_type())
```

struct AgsAudioUnitPort

```
struct AgsAudioUnitPort;
```

struct AgsAudioUnitPortClass

```
struct AgsAudioUnitPortClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-unit-client" property

```
  "audio-unit-client"         AgsAudioUnitClient~*
```

The assigned AgsAudioUnitClient.

Owner: AgsAudioUnitPort

Flags: Read / Write

Since: 3.0.0

The "audio-unit-device" property

```
  "audio-unit-device"         GObject~*
```

The assigned AgsAudioUnitDevout.

Owner: AgsAudioUnitPort

Flags: Read / Write

Since: 3.0.0

The "port-name" property

```
"port-name"                 char~*
```

The core audio soundcard indentifier

Owner: AgsAudioUnitPort

Flags: Read / Write

Default value: "hw:0"

Since: 3.0.0

8.35 AgsAudioUnitDevin

AgsAudioUnitDevin — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_UNIT_DEVIN_GET_OBJ_MUTEX()
GQuark	ags_audio_unit_devin_error_quark ()
gboolean	ags_audio_unit_devin_test_flags ()
void	ags_audio_unit_devin_set_flags ()
void	ags_audio_unit_devin_unset_flags ()
void	ags_audio_unit_devin_switch_buffer_flag ()
void	ags_audio_unit_devin_adjust_delay_and_attack ()
void	ags_audio_unit_devin_realloc_buffer ()
AgsAudioUnitDevin *	ags_audio_unit_devin_new ()
#define	AGS_AUDIO_UNIT_DEVIN()
#define	AGS_AUDIO_UNIT_DEVIN_CLASS()
#define	AGS_AUDIO_UNIT_DEVIN_GET_CLASS()
#define	AGS_IS_AUDIO_UNIT_DEVIN()
#define	AGS_IS_AUDIO_UNIT_DEVIN_CLASS()
GType	ags_audio_unit_devin_flags_get_type ()
GType	ags_audio_unit_devin_get_type ()

Properties

gpointer	attack	Read
AgsAudioUnitClient *	audio-unit-client	Read / Write
gpointer	audio-unit-port	Read / Write
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write

guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_AUDIO_UNIT_DEVIN_DEFAULT_APP_BUFFER_SIZE
enum	AgsAudioUnitDevinFlags
enum	AgsAudioUnitDevinAppBufferMode
enum	AgsAudioUnitDevinSyncFlags
#define	AGS_AUDIO_UNIT_DEVIN_ERROR
enum	AgsAudioUnitDevinError
#define	AGS_TYPE_AUDIO_UNIT_DEVIN
struct	AgsAudioUnitDevin
struct	AgsAudioUnitDevinClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsAudioUnitDevinFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsAudioUnitDevin
```

Implemented Interfaces

AgsAudioUnitDevin implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/audio-unit/ags_audio_unit_devin.h>
```

Description

AgsAudioUnitDevin represents a soundcard and supports output.

Functions

AGS_AUDIO_UNIT_DEVIN_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_UNIT_DEVIN_GET_OBJ_MUTEX(obj) (&(((AgsAudioUnitDevin *) obj)->obj_mutex))
```

ags_audio_unit_devin_error_quark ()

```
GQuark
ags_audio_unit_devin_error_quark ();
```

ags_audio_unit_devin_test_flags ()

```
gboolean
ags_audio_unit_devin_test_flags (AgsAudioUnitDevin *audio_unit_devin,
                                 guint flags);
```

Test *flags* to be set on *audio_unit_devin* .

Parameters

audio_unit_devin	the AgsAudioUnitDevin	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_unit_devin_set_flags ()

```
void
ags_audio_unit_devin_set_flags (AgsAudioUnitDevin *audio_unit_devin,
                                guint flags);
```

Enable a feature of *audio_unit_devin* .

Parameters

audio_unit_devin	the AgsAudioUnitDevin	
flags	see AgsAudioUnitDevinFlags	

Since: 3.0.0

ags_audio_unit_devin_unset_flags ()

```
void
ags_audio_unit_devin_unset_flags (AgsAudioUnitDevin *audio_unit_devin,
                                  guint flags);
```

Disable a feature of *audio_unit_devin* .

Parameters

audio_unit_devin	the AgsAudioUnitDevin	
flags	see AgsAudioUnitDevinFlags	

Since: 3.0.0

ags_audio_unit_devin_switch_buffer_flag ()

```
void
ags_audio_unit_devin_switch_buffer_flag
                                (AgsAudioUnitDevin *audio_unit_devin);
```

The buffer flag indicates the currently played buffer.

Parameters

audio_unit_devin	an AgsAudioUnitDevin

Since: 3.0.0

ags_audio_unit_devin_adjust_delay_and_attack ()

```
void
ags_audio_unit_devin_adjust_delay_and_attack
                                (AgsAudioUnitDevin *audio_unit_devin);
```

Calculate delay and attack and reset it.

Parameters

audio_unit_devin	the AgsAudioUnitDevin

Since: 3.0.0

ags_audio_unit_devin_realloc_buffer ()

```
void
ags_audio_unit_devin_realloc_buffer (AgsAudioUnitDevin *audio_unit_devin);
```

Reallocate the internal audio buffer.

Parameters

audio_unit_devin	the AgsAudioUnitDevin

Since: 3.0.0

ags_audio_unit_devin_new ()

```
AgsAudioUnitDevin~*
ags_audio_unit_devin_new ();
```

Creates a new instance of AgsAudioUnitDevin.

Returns

a new AgsAudioUnitDevin

Since: 3.0.0

AGS_AUDIO_UNIT_DEVIN()

```
#define AGS_AUDIO_UNIT_DEVIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_DEVIN, AgsAudioUnitDevin))
```

AGS_AUDIO_UNIT_DEVIN_CLASS()

```
#define AGS_AUDIO_UNIT_DEVIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_UNIT_DEVIN, AgsAudioUnitDevin))
```

AGS_AUDIO_UNIT_DEVIN_GET_CLASS()

```
#define AGS_AUDIO_UNIT_DEVIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_UNIT_DEVIN, AgsAudioUnitDevinClass))
```

AGS_IS_AUDIO_UNIT_DEVIN()

```
#define AGS_IS_AUDIO_UNIT_DEVIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_DEVIN))
```

AGS_IS_AUDIO_UNIT_DEVIN_CLASS()

```
#define AGS_IS_AUDIO_UNIT_DEVIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_UNIT_DEVIN))
```

ags_audio_unit_devin_flags_get_type ()

```
GType
ags_audio_unit_devin_flags_get_type ();
```

ags_audio_unit_devin_get_type ()

```
GType
ags_audio_unit_devin_get_type ();
```

Types and Values

AGS_AUDIO_UNIT_DEVIN_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_AUDIO_UNIT_DEVIN_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsAudioUnitDevinFlags

Enum values to control the behavior or indicate internal state of AgsAudioUnitDevin by enable/disable as flags.

Members

AGS_AUDIO_UNIT_DEVIN_INITIALIZED	the sound-card was ini-tial-ized
AGS_AUDIO_UNIT_DEVIN_START_RECORD	capture start-ing
AGS_AUDIO_UNIT_DEVIN_RECORD	do cap-ture
AGS_AUDIO_UNIT_DEVIN_SHUTDOWN	stop cap-ture
AGS_AUDIO_UNIT_DEVIN_NONBLOCKING	do non-blocking calls
AGS_AUDIO_UNIT_DEVIN_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsAudioUnitDevinAppBufferMode

Enum values to indicate internal state of AgsAudioUnitDevin application buffer by setting mode.

Members

AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_0	ring-buffer 0
AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_1	ring-buffer 1
AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_2	ring-buffer 2

AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_3	ring-buffer 3
AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_4	ring-buffer 4
AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_5	ring-buffer 5
AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_6	ring-buffer 6
AGS_AUDIO_UNIT_DEVIN_APP_BUFFER_7	ring-buffer 7

enum AgsAudioUnitDevinSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_AUDIO_UNIT_DEVIN_PASS_THROUGH	do not sync
AGS_AUDIO_UNIT_DEVIN_INITIAL_CALLBACK	initial call-back
AGS_AUDIO_UNIT_DEVIN_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_AUDIO_UNIT_DEVIN_CALLBACK_DONE	sync done, sound-card con-di-tional lock
AGS_AUDIO_UNIT_DEVIN_CALLBACK_FINISH_WAIT	sync wait, client con-di-tional lock

AGS_AUDIO_UNIT_DEVIN_CALLBACK_FINISH_DONE	sync done, client conditional lock

AGS_AUDIO_UNIT_DEVIN_ERROR

```
#define AGS_AUDIO_UNIT_DEVIN_ERROR (ags_audio_unit_devin_error_quark())
```

enum AgsAudioUnitDevinError

Members

AGS_AUDIO_UNIT_DEVIN_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_AUDIO_UNIT_DEVIN

```
#define AGS_TYPE_AUDIO_UNIT_DEVIN                (ags_audio_unit_devin_get_type())
```

struct AgsAudioUnitDevin

```
struct AgsAudioUnitDevin;
```

struct AgsAudioUnitDevinClass

```
struct AgsAudioUnitDevinClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsAudioUnitDevin

Flags: Read

Since: 3.0.0

The "audio-unit-client" property

```
"audio-unit-client"        AgsAudioUnitClient~*
```

The assigned AgsAudioUnitClient

Owner: AgsAudioUnitDevin

Flags: Read / Write

Since: 3.0.0

The "audio-unit-port" property

```
"audio-unit-port"          gpointer
```

The assigned AgsAudioUnitPort

Owner: AgsAudioUnitDevin

Flags: Read / Write

Since: 3.0.0

The "bpm" property

```
"bpm"                      double
```

Beats per minute

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
"buffer"                   gpointer
```

The buffer

Owner: AgsAudioUnitDevin

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"              guint
```

The buffer size

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [1,44100]

Default value: 940

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The audio unit soundcard indentifier

Owner: AgsAudioUnitDevin

Flags: Read / Write

Default value: "ags-audio-unit-devin-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsAudioUnitDevin

Flags: Read / Write

Allowed values: [8000,96000]

Default value: 44100

Since: 3.0.0

8.36 AgsAudioUnitDevout

AgsAudioUnitDevout — Output to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_UNIT_DEVOUT_GET_OBJ_MUTEX()
GQuark	ags_audio_unit_devout_error_quark ()
gboolean	ags_audio_unit_devout_test_flags ()
void	ags_audio_unit_devout_set_flags ()
void	ags_audio_unit_devout_unset_flags ()
void	ags_audio_unit_devout_switch_buffer_flag ()
void	ags_audio_unit_devout_adjust_delay_and_attack ()
void	ags_audio_unit_devout_realloc_buffer ()
AgsAudioUnitDevout *	ags_audio_unit_devout_new ()
#define	AGS_AUDIO_UNIT_DEVOUT()
#define	AGS_AUDIO_UNIT_DEVOUT_CLASS()
#define	AGS_AUDIO_UNIT_DEVOUT_GET_CLASS()
#define	AGS_IS_AUDIO_UNIT_DEVOUT()
#define	AGS_IS_AUDIO_UNIT_DEVOUT_CLASS()

GType	ags_audio_unit_devout_flags_get_type ()
GType	ags_audio_unit_devout_get_type ()

Properties

gpointer	attack	Read
AgsAudioUnitClient *	audio-unit-client	Read / Write
gpointer	audio-unit-port	Read / Write
double	bpm	Read / Write
gpointer	buffer	Read
guint	buffer-size	Read / Write
double	delay-factor	Read / Write
char *	device	Read / Write
guint	dsp-channels	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_AUDIO_UNIT_DEVOUT_DEFAULT_APP_BUFFER_SIZE
enum	AgsAudioUnitDevoutFlags
enum	AgsAudioUnitDevoutAppBufferMode
enum	AgsAudioUnitDevoutSyncFlags
#define	AGS_AUDIO_UNIT_DEVOUT_ERROR
enum	AgsAudioUnitDevoutError
#define	AGS_TYPE_AUDIO_UNIT_DEVOUT
struct	AgsAudioUnitDevout
struct	AgsAudioUnitDevoutClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsAudioUnitDevoutFlags
GObject
&#x2570;&#x2500;&#x2500; AgsAudioUnitDevout
```

Implemented Interfaces

AgsAudioUnitDevout implements AgsConnectable and AgsSoundcard.

Includes

```
#include <ags/audio/audio-unit/ags_audio_unit_devout.h>
```

Description

AgsAudioUnitDevout represents a soundcard and supports output.

Functions

AGS_AUDIO_UNIT_DEVOUT_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_UNIT_DEVOUT_GET_OBJ_MUTEX(obj) (&(((AgsAudioUnitDevout *) obj)->obj_mutex ↩
    ))
```

ags_audio_unit_devout_error_quark ()

```
GQuark
ags_audio_unit_devout_error_quark ();
```

ags_audio_unit_devout_test_flags ()

```
gboolean
ags_audio_unit_devout_test_flags (AgsAudioUnitDevout *audio_unit_devout,
                                  guint flags);
```

Test *flags* to be set on *audio_unit_devout* .

Parameters

audio_unit_devout	the AgsAudioUnitDevout	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_unit_devout_set_flags ()

```
void
ags_audio_unit_devout_set_flags (AgsAudioUnitDevout *audio_unit_devout,
                                 guint flags);
```

Enable a feature of *audio_unit_devout* .

Parameters

audio_unit_devout	the AgsAudioUnitDevout	
flags	see AgsAudioUnitDevoutFlags	

Since: 3.0.0

ags_audio_unit_devout_unset_flags ()

```
void
ags_audio_unit_devout_unset_flags (AgsAudioUnitDevout *audio_unit_devout,
                                   guint flags);
```

Disable a feature of *audio_unit_devout* .

Parameters

audio_unit_devout	the AgsAudioUnitDevout	
flags	see AgsAudioUnitDevoutFlags	

Since: 3.0.0

ags_audio_unit_devout_switch_buffer_flag ()

```
void
ags_audio_unit_devout_switch_buffer_flag
                                (AgsAudioUnitDevout *audio_unit_devout);
```

The buffer flag indicates the currently played buffer.

Parameters

audio_unit_devout	an AgsAudioUnitDevout	

Since: 3.0.0

ags_audio_unit_devout_adjust_delay_and_attack ()

```
void
ags_audio_unit_devout_adjust_delay_and_attack
                                (AgsAudioUnitDevout *audio_unit_devout);
```

Calculate delay and attack and reset it.

Parameters

audio_unit_devout	the AgsAudioUnitDevout	

Since: 3.0.0

ags_audio_unit_devout_realloc_buffer ()

```
void
ags_audio_unit_devout_realloc_buffer (AgsAudioUnitDevout *audio_unit_devout);
```

Reallocate the internal audio buffer.

Parameters

audio_unit_devout	the AgsAudioUnitDevout

Since: 3.0.0

ags_audio_unit_devout_new ()

```
AgsAudioUnitDevout~*
ags_audio_unit_devout_new ();
```

Creates a new instance of AgsAudioUnitDevout.

Returns

a new AgsAudioUnitDevout

Since: 3.0.0

AGS_AUDIO_UNIT_DEVOUT()

```
#define AGS_AUDIO_UNIT_DEVOUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_DEVOUT, AgsAudioUnitDevout))
```

AGS_AUDIO_UNIT_DEVOUT_CLASS()

```
#define AGS_AUDIO_UNIT_DEVOUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_UNIT_DEVOUT, AgsAudioUnitDevout))
```

AGS_AUDIO_UNIT_DEVOUT_GET_CLASS()

```
#define AGS_AUDIO_UNIT_DEVOUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_UNIT_DEVOUT, AgsAudioUnitDevoutClass))
```

AGS_IS_AUDIO_UNIT_DEVOUT()

```
#define AGS_IS_AUDIO_UNIT_DEVOUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_UNIT_DEVOUT))
```

AGS_IS_AUDIO_UNIT_DEVOUT_CLASS()

```
#define AGS_IS_AUDIO_UNIT_DEVOUT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_UNIT_DEVOUT))
```

ags_audio_unit_devout_flags_get_type ()

```
GType
ags_audio_unit_devout_flags_get_type ();
```

ags_audio_unit_devout_get_type ()

```
GType
ags_audio_unit_devout_get_type ();
```

Types and Values

AGS_AUDIO_UNIT_DEVOUT_DEFAULT_APP_BUFFER_SIZE

```
#define AGS_AUDIO_UNIT_DEVOUT_DEFAULT_APP_BUFFER_SIZE (8)
```

enum AgsAudioUnitDevoutFlags

Enum values to control the behavior or indicate internal state of AgsAudioUnitDevout by enable/disable as flags.

Members

AGS_AUDIO_UNIT_DEVOUT_INITIALIZED	the sound-card was ini-tial-ized
AGS_AUDIO_UNIT_DEVOUT_START_PLAY	playback start-ing
AGS_AUDIO_UNIT_DEVOUT_PLAY	do play-back
AGS_AUDIO_UNIT_DEVOUT_SHUTDOWN	stop play-back
AGS_AUDIO_UNIT_DEVOUT_NONBLOCKING	do non-blocking calls
AGS_AUDIO_UNIT_DEVOUT_ATTACK_FIRST	use first at-tack, in-stead of sec-ond one

enum AgsAudioUnitDevoutAppBufferMode

Enum values to indicate internal state of AgsAudioUnitDevout application buffer by setting mode.

Members

AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_0	ring-buffer 0
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_1	ring-buffer 1
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_2	ring-buffer 2
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_3	ring-buffer 3
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_4	ring-buffer 4
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_5	ring-buffer 5
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_6	ring-buffer 6
AGS_AUDIO_UNIT_DEVOUT_APP_BUFFER_7	ring-buffer 7

enum AgsAudioUnitDevoutSyncFlags

Enum values to control the synchronization between soundcard and client.

Members

AGS_AUDIO_UNIT_DEVOUT_PASS_THROUGH	do not sync
AGS_AUDIO_UNIT_DEVOUT_INITIAL_CALLBACK	initial call-back
AGS_AUDIO_UNIT_DEVOUT_CALLBACK_WAIT	sync wait, sound-card con-di-tional lock
AGS_AUDIO_UNIT_DEVOUT_CALLBACK_DONE	sync done, sound-card con-di-tional lock

AGS_AUDIO_UNIT_DEVOUT_CALLBACK_FINISH_WAIT	sync wait, client conditional lock
AGS_AUDIO_UNIT_DEVOUT_CALLBACK_FINISH_DONE	sync done, client conditional lock

AGS_AUDIO_UNIT_DEVOUT_ERROR

```
#define AGS_AUDIO_UNIT_DEVOUT_ERROR (ags_audio_unit_devout_error_quark())
```

enum AgsAudioUnitDevoutError

Members

AGS_AUDIO_UNIT_DEVOUT_ERROR_LOCKED_SOUNDCARD	

AGS_TYPE_AUDIO_UNIT_DEVOUT

```
#define AGS_TYPE_AUDIO_UNIT_DEVOUT                (ags_audio_unit_devout_get_type())
```

struct AgsAudioUnitDevout

```
struct AgsAudioUnitDevout;
```

struct AgsAudioUnitDevoutClass

```
struct AgsAudioUnitDevoutClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    gpointer
```

Attack of the buffer

Owner: AgsAudioUnitDevout

Flags: Read

Since: 3.0.0

The "audio-unit-client" property

```
"audio-unit-client"          AgsAudioUnitClient~*
```

The assigned AgsAudioUnitClient

Owner: AgsAudioUnitDevout

Flags: Read / Write

Since: 3.0.0

The "audio-unit-port" property

```
"audio-unit-port"            gpointer
```

The assigned AgsAudioUnitPort

Owner: AgsAudioUnitDevout

Flags: Read / Write

Since: 3.0.0

The "bpm" property

```
"bpm"                        double
```

Beats per minute

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [1,240]

Default value: 120

Since: 3.0.0

The "buffer" property

```
"buffer"                     gpointer
```

The buffer

Owner: AgsAudioUnitDevout

Flags: Read

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [1,44100]

Default value: 940

Since: 3.0.0

The "delay-factor" property

```
"delay-factor"              double
```

tact

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [0,16]

Default value: 1

Since: 3.0.0

The "device" property

```
"device"                    char~*
```

The audio unit soundcard indentifier

Owner: AgsAudioUnitDevout

Flags: Read / Write

Default value: "ags-audio-unit-devout-0"

Since: 3.0.0

The "dsp-channels" property

```
"dsp-channels"              guint
```

The dsp channel count

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The precision of the buffer

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 16

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The pcm channel count

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [1,64]

Default value: 2

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate

Owner: AgsAudioUnitDevout

Flags: Read / Write

Allowed values: [8000,96000]

Default value: 44100

Since: 3.0.0

Chapter 9

The audio tree

The audio tree does organize your data in a hierarchical structure.

9.1 AgsSoundProvider

AgsSoundProvider — Sound provider interface

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_sound_provider_set_default_soundcard ()
GObject *	ags_sound_provider_get_default_soundcard ()
void	ags_sound_provider_set_default_soundcard_thread ()
GObject *	ags_sound_provider_get_default_soundcard_thread ()
void	ags_sound_provider_set_soundcard ()
GList *	ags_sound_provider_get_soundcard ()
void	ags_sound_provider_set_sequencer ()
GList *	ags_sound_provider_get_sequencer ()
void	ags_sound_provider_set_audio ()
GList *	ags_sound_provider_get_audio ()
void	ags_sound_provider_set_sound_server ()
GList *	ags_sound_provider_get_sound_server ()
void	ags_sound_provider_set_osc_server ()
GList *	ags_sound_provider_get_osc_server ()
#define	AGS_IS_SOUND_PROVIDER()
#define	AGS_IS_SOUND_PROVIDER_INTERFACE()
#define	AGS_SOUND_PROVIDER()
#define	AGS_SOUND_PROVIDER_GET_INTERFACE()
#define	AGS_SOUND_PROVIDER_INTERFACE()
GType	ags_sound_provider_get_type ()

Types and Values

#define	AGS_TYPE_SOUND_PROVIDER
	AgsSoundProvider
struct	AgsSoundProviderInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsSoundProvider
```

Known Implementations

AgsSoundProvider is implemented by AgsAudioApplicationContext.

Includes

```
#include <ags/audio/ags_sound_provider.h>
```

Description

The AgsSoundProvider provides you the advanced audio capabilities.

Functions

ags_sound_provider_set_default_soundcard ()

```
void
ags_sound_provider_set_default_soundcard
                                (AgsSoundProvider *sound_provider,
                                 GObject *soundcard);
```

Set default soundcard.

Parameters

sound_provider	the AgsSoundProvider	
soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_sound_provider_get_default_soundcard ()

```
GObject~*
ags_sound_provider_get_default_soundcard
                                (AgsSoundProvider *sound_provider);
```

Get default soundcard thread.

Parameters

sound_provider	the AgsSoundProvider

Returns

the GObject implementing AgsSoundcard.

[transfer full]

Since: 3.0.0

ags_sound_provider_set_default_soundcard_thread ()

```
void
ags_sound_provider_set_default_soundcard_thread
                                (AgsSoundProvider *sound_provider,
                                 GObject *soundcard_thread);
```

Set default soundcard thread.

Parameters

sound_provider	the AgsSoundProvider
soundcard_thread	the default AgsThread

Since: 3.0.0

ags_sound_provider_get_default_soundcard_thread ()

```
GObject~*
ags_sound_provider_get_default_soundcard_thread
                                (AgsSoundProvider *sound_provider);
```

Get default soundcard thread.

Parameters

sound_provider	the AgsSoundProvider

Returns

the AgsThread.

[transfer full]

Since: 3.0.0

ags_sound_provider_set_soundcard ()

```
void
ags_sound_provider_set_soundcard (AgsSoundProvider *sound_provider,
                                  GList *soundcard);
```

Set soundcards.

Parameters

sound_provider	the AgsSoundProvider	
soundcard	a GList containing AgsSoundcard.	*[element-type GObject][transfer full]*

Since: 3.0.0

ags_sound_provider_get_soundcard ()

```
GList~*
ags_sound_provider_get_soundcard (AgsSoundProvider *sound_provider);
```

Get soundcards.

Parameters

sound_provider	the AgsSoundProvider	

Returns

a GList containing AgsSoundcard.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_provider_set_sequencer ()

```
void
ags_sound_provider_set_sequencer (AgsSoundProvider *sound_provider,
                                  GList *sequencer);
```

Set sequencers.

Parameters

sound_provider	the AgsSoundProvider	
sequencer	a GList containing AgsSequencer.	*[element-type GObject][transfer full]*

Since: 3.0.0

ags_sound_provider_get_sequencer ()

```
GList~*
ags_sound_provider_get_sequencer (AgsSoundProvider *sound_provider);
```

Get sequencers.

Parameters

sound_provider	the AgsSoundProvider

Returns

a GList containing AgsSequencer.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_provider_set_audio ()

```
void
ags_sound_provider_set_audio (AgsSoundProvider *sound_provider,
                              GList *audio);
```

Set audio objects.

Parameters

sound_provider	the AgsSoundProvider	
audio	a GList containing AgsAudio.	*[element-type AgsAudio.Audio][transfer full]*

Since: 3.0.0

ags_sound_provider_get_audio ()

```
GList~*
ags_sound_provider_get_audio (AgsSoundProvider *sound_provider);
```

Get audio objects.

Parameters

sound_provider	the AgsSoundProvider

Returns

a GList containing AgsAudio.

[element-type AgsAudio.Audio][transfer full]

Since: 3.0.0

ags_sound_provider_set_sound_server ()

```
void
ags_sound_provider_set_sound_server (AgsSoundProvider *sound_provider,
                                     GList *sound_server);
```

Get sound server.

Parameters

sound_provider	the AgsSoundProvider	
sound_server	a GList containing AgsSoundServer.	*[element-type GObject][transfer full]*

Since: 3.0.0

ags_sound_provider_get_sound_server ()

```
GList~*
ags_sound_provider_get_sound_server (AgsSoundProvider *sound_provider);
```

Set sound server.

Parameters

sound_provider	the AgsSoundProvider	

Returns

a GList containing AgsSoundServer.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_provider_set_osc_server ()

```
void
ags_sound_provider_set_osc_server (AgsSoundProvider *sound_provider,
                                   GList *osc_server);
```

Set OSC server.

Parameters

sound_provider	the AgsSoundProvider	
osc_server	a GList containing AgsOscServer.	*[element-type GObject][transfer full]*

Since: 3.0.0

ags_sound_provider_get_osc_server ()

```
GList~*
ags_sound_provider_get_osc_server (AgsSoundProvider *sound_provider);
```

Set distributed manager.

Parameters

sound_provider	the AgsSoundProvider

Returns

a GList containing AgsOscServer.

[element-type GObject][transfer full]

Since: 3.0.0

AGS_IS_SOUND_PROVIDER()

```
#define AGS_IS_SOUND_PROVIDER(obj)                  (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SOUND_PROVIDER))
```

AGS_IS_SOUND_PROVIDER_INTERFACE()

```
#define AGS_IS_SOUND_PROVIDER_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SOUND_PROVIDER))
```

AGS_SOUND_PROVIDER()

```
#define AGS_SOUND_PROVIDER(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOUND_PROVIDER, AgsSoundProvider))
```

AGS_SOUND_PROVIDER_GET_INTERFACE()

```
#define AGS_SOUND_PROVIDER_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SOUND_PROVIDER, AgsSoundProviderInterface))
```

AGS_SOUND_PROVIDER_INTERFACE()

```
#define AGS_SOUND_PROVIDER_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SOUND_PROVIDER, AgsSoundProviderInterface))
```

ags_sound_provider_get_type ()

```
GType
ags_sound_provider_get_type ();
```

Types and Values

AGS_TYPE_SOUND_PROVIDER

```
#define AGS_TYPE_SOUND_PROVIDER                     (ags_sound_provider_get_type())
```

AgsSoundProvider

```
typedef struct _AgsSoundProvider AgsSoundProvider;
```

struct AgsSoundProviderInterface

```
struct AgsSoundProviderInterface {
  GTypeInterface ginterface;

  void (*set_default_soundcard)(AgsSoundProvider *sound_provider,
        GObject *soundcard);
  GObject* (*get_default_soundcard)(AgsSoundProvider *sound_provider);

  void (*set_default_soundcard_thread)(AgsSoundProvider *sound_provider,
              GObject *soundcard_thread);
  GObject* (*get_default_soundcard_thread)(AgsSoundProvider *sound_provider);

  void (*set_soundcard)(AgsSoundProvider *sound_provider,
      GList *soundcard);
  GList* (*get_soundcard)(AgsSoundProvider *sound_provider);

  void (*set_sequencer)(AgsSoundProvider *sound_provider,
      GList *sequencer);
  GList* (*get_sequencer)(AgsSoundProvider *sound_provider);

  void (*set_audio)(AgsSoundProvider *sound_provider,
        GList *audio);
  GList* (*get_audio)(AgsSoundProvider *sound_provider);

  void (*set_sound_server)(AgsSoundProvider *sound_provider,
         GList *sound_server);
  GList* (*get_sound_server)(AgsSoundProvider *sound_provider);

  void (*set_osc_server)(AgsSoundProvider *sound_provider,
       GList *osc_server);
  GList* (*get_osc_server)(AgsSoundProvider *sound_provider);
};
```

9.2 AgsAudioApplicationContext

AgsAudioApplicationContext — audio application context

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_audio_application_context_loader_timeout ()
AgsAudioApplicationContext *	ags_audio_application_context_new ()
#define	AGS_AUDIO_APPLICATION_CONTEXT()
#define	AGS_AUDIO_APPLICATION_CONTEXT_CLASS()
#define	AGS_AUDIO_APPLICATION_CONTEXT_GET_CLASS()

#define	AGS_IS_AUDIO_APPLICATION_CONTEXT()
#define	AGS_IS_AUDIO_APPLICATION_CONTEXT_CLASS()
GType	ags_audio_application_context_get_type ()

Types and Values

#define	AGS_AUDIO_DEFAULT_VERSION
#define	AGS_AUDIO_BUILD_ID
#define	AGS_EFFECTS_DEFAULT_VERSION
#define	AGS_AUDIO_APPLICATION_CONTEXT_DEFAULT_LOADER_INTERVAL
#define	AGS_TYPE_AUDIO_APPLICATION_CONTEXT
struct	AgsAudioApplicationContext
struct	AgsAudioApplicationContextClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsApplicationContext
        &#x2570;&#x2500;&#x2500; AgsAudioApplicationContext
```

Implemented Interfaces

AgsAudioApplicationContext implements AgsConnectable, AgsConcurrencyProvider, AgsServiceProvider and AgsSoundProvider.

Includes

```
#include <ags/audio/ags_audio_application_context.h>
```

Description

The AgsAudioApplicationContext provides you sound processing, output and capturing.

Functions

ags_audio_application_context_loader_timeout ()

```
gboolean
ags_audio_application_context_loader_timeout
                                (AgsAudioApplicationContext *audio_application_context);
```

ags_audio_application_context_new ()

```
AgsAudioApplicationContext~*
ags_audio_application_context_new ();
```

AGS_AUDIO_APPLICATION_CONTEXT()

```
#define AGS_AUDIO_APPLICATION_CONTEXT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_AUDIO_APPLICATION_CONTEXT, AgsAudioApplicationContext))
```

AGS_AUDIO_APPLICATION_CONTEXT_CLASS()

```
#define AGS_AUDIO_APPLICATION_CONTEXT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_APPLICATION_CONTEXT, AgsAudioApplicationContextClass))
```

AGS_AUDIO_APPLICATION_CONTEXT_GET_CLASS()

```
#define AGS_AUDIO_APPLICATION_CONTEXT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_APPLICATION_CONTEXT, AgsAudioApplicationContextClass))
```

AGS_IS_AUDIO_APPLICATION_CONTEXT()

```
#define AGS_IS_AUDIO_APPLICATION_CONTEXT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_AUDIO_APPLICATION_CONTEXT))
```

AGS_IS_AUDIO_APPLICATION_CONTEXT_CLASS()

```
#define AGS_IS_AUDIO_APPLICATION_CONTEXT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_AUDIO_APPLICATION_CONTEXT))
```

ags_audio_application_context_get_type ()

```
GType
ags_audio_application_context_get_type
                                ();
```

Types and Values

AGS_AUDIO_DEFAULT_VERSION

```
#define AGS_AUDIO_DEFAULT_VERSION "3.0.0"
```

AGS_AUDIO_BUILD_ID

```
#define AGS_AUDIO_BUILD_ID "Fri Nov  8 21:47:01 UTC 2019"
```

AGS_EFFECTS_DEFAULT_VERSION

```
#define AGS_EFFECTS_DEFAULT_VERSION "0.7.13"
```

AGS_AUDIO_APPLICATION_CONTEXT_DEFAULT_LOADER_INTERVAL

```
#define AGS_AUDIO_APPLICATION_CONTEXT_DEFAULT_LOADER_INTERVAL (1000 / 25)
```

AGS_TYPE_AUDIO_APPLICATION_CONTEXT

```
#define AGS_TYPE_AUDIO_APPLICATION_CONTEXT                ( ↩
    ags_audio_application_context_get_type())
```

struct AgsAudioApplicationContext

```
struct AgsAudioApplicationContext;
```

struct AgsAudioApplicationContextClass

```
struct AgsAudioApplicationContextClass {
  AgsApplicationContextClass application_context;
};
```

9.3 AgsAudio

AgsAudio — A container of channels organizing them as input or output

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_GET_OBJ_MUTEX()
#define	AGS_AUDIO_GET_PLAY_MUTEX()
#define	AGS_AUDIO_GET_RECALL_MUTEX()
GRecMutex *	ags_audio_get_obj_mutex ()
GRecMutex *	ags_audio_get_play_mutex ()
GRecMutex *	ags_audio_get_recall_mutex ()
gboolean	ags_audio_test_flags ()
void	ags_audio_set_flags ()
void	ags_audio_unset_flags ()
gboolean	ags_audio_test_ability_flags ()
void	ags_audio_set_ability_flags ()
void	ags_audio_unset_ability_flags ()
gboolean	ags_audio_test_behaviour_flags ()
void	ags_audio_set_behaviour_flags ()
void	ags_audio_unset_behaviour_flags ()
gboolean	ags_audio_test_staging_flags ()
void	ags_audio_set_staging_flags ()
void	ags_audio_unset_staging_flags ()
gboolean	ags_audio_test_staging_completed ()
void	ags_audio_set_staging_completed ()
void	ags_audio_unset_staging_completed ()
gchar *	ags_audio_get_audio_name ()
void	ags_audio_set_audio_name ()
GList *	ags_audio_find_name ()

guint	ags_audio_get_max_audio_channels ()
void	ags_audio_set_max_audio_channels ()
guint	ags_audio_get_max_output_pads ()
void	ags_audio_set_max_output_pads ()
guint	ags_audio_get_max_input_pads ()
void	ags_audio_set_max_input_pads ()
void	ags_audio_set_max_pads ()
guint	ags_audio_get_audio_channels ()
void	ags_audio_set_audio_channels ()
guint	ags_audio_get_output_pads ()
void	ags_audio_set_output_pads ()
guint	ags_audio_get_input_pads ()
void	ags_audio_set_input_pads ()
void	ags_audio_set_pads ()
guint	ags_audio_get_output_lines ()
guint	ags_audio_get_input_lines ()
AgsChannel *	ags_audio_get_output ()
AgsChannel *	ags_audio_get_input ()
GObject *	ags_audio_get_output_soundcard ()
void	ags_audio_set_output_soundcard ()
GObject *	ags_audio_get_input_soundcard ()
void	ags_audio_set_input_soundcard ()
GObject *	ags_audio_get_output_sequencer ()
void	ags_audio_set_output_sequencer ()
GObject *	ags_audio_get_input_sequencer ()
void	ags_audio_set_input_sequencer ()
guint	ags_audio_get_samplerate ()
void	ags_audio_set_samplerate ()
guint	ags_audio_get_buffer_size ()
void	ags_audio_set_buffer_size ()
guint	ags_audio_get_format ()
void	ags_audio_set_format ()
gdouble	ags_audio_get_bpm ()
void	ags_audio_set_bpm ()
guint	ags_audio_get_audio_start_mapping ()
void	ags_audio_set_audio_start_mapping ()
guint	ags_audio_get_midi_start_mapping ()
void	ags_audio_set_midi_start_mapping ()
guint	ags_audio_get_midi_channel ()
void	ags_audio_set_midi_channel ()
guint	ags_audio_get_numerator ()
void	ags_audio_set_numerator ()
guint	ags_audio_get_denominator ()
void	ags_audio_set_denominator ()
gchar *	ags_audio_get_time_signature ()
void	ags_audio_set_time_signature ()
gboolean	ags_audio_get_is_minor ()
void	ags_audio_set_is_minor ()
guint	ags_audio_get_sharp_flats ()
void	ags_audio_set_sharp_flats ()
gint	ags_audio_get_octave ()
void	ags_audio_set_octave ()
guint	ags_audio_get_key ()
void	ags_audio_set_key ()
gint	ags_audio_get_absolute_key ()
void	ags_audio_set_absolute_key ()
GList *	ags_audio_get_preset ()

void	ags_audio_set_preset ()
void	ags_audio_add_preset ()
void	ags_audio_remove_preset ()
GObject *	ags_audio_get_playback_domain ()
void	ags_audio_set_playback_domain ()
GList *	ags_audio_get_synth_generator ()
void	ags_audio_set_synth_generator ()
void	ags_audio_add_synth_generator ()
void	ags_audio_remove_synth_generator ()
GList *	ags_audio_get_sf2_synth_generator ()
void	ags_audio_set_sf2_synth_generator ()
void	ags_audio_add_sf2_synth_generator ()
void	ags_audio_remove_sf2_synth_generator ()
GList *	ags_audio_get_sfz_synth_generator ()
void	ags_audio_set_sfz_synth_generator ()
void	ags_audio_add_sfz_synth_generator ()
void	ags_audio_remove_sfz_synth_generator ()
GList *	ags_audio_get_cursor ()
void	ags_audio_set_cursor ()
void	ags_audio_add_cursor ()
void	ags_audio_remove_cursor ()
GList *	ags_audio_get_notation ()
void	ags_audio_set_notation ()
void	ags_audio_add_notation ()
void	ags_audio_remove_notation ()
gchar **	ags_audio_get_automation_port ()
void	ags_audio_set_automation_port ()
void	ags_audio_add_automation_port ()
void	ags_audio_remove_automation_port ()
GList *	ags_audio_get_automation ()
void	ags_audio_set_automation ()
void	ags_audio_add_automation ()
void	ags_audio_remove_automation ()
GList *	ags_audio_get_wave ()
void	ags_audio_set_wave ()
void	ags_audio_add_wave ()
void	ags_audio_remove_wave ()
GObject *	ags_audio_get_output_audio_file ()
void	ags_audio_set_output_audio_file ()
GObject *	ags_audio_get_input_audio_file ()
void	ags_audio_set_input_audio_file ()
gchar *	ags_audio_get_instrument_name ()
void	ags_audio_set_instrument_name ()
gchar *	ags_audio_get_track_name ()
void	ags_audio_set_track_name ()
GList *	ags_audio_get_midi ()
void	ags_audio_set_midi ()
void	ags_audio_add_midi ()
void	ags_audio_remove_midi ()
GObject *	ags_audio_get_output_midi_file ()
void	ags_audio_set_output_midi_file ()
GObject *	ags_audio_get_input_midi_file ()
void	ags_audio_set_input_midi_file ()
GList *	ags_audio_get_recall_id ()
void	ags_audio_set_recall_id ()
void	ags_audio_add_recall_id ()

void	ags_audio_remove_recall_id ()
GList *	ags_audio_get_recycling_context ()
void	ags_audio_set_recycling_context ()
void	ags_audio_add_recycling_context ()
void	ags_audio_remove_recycling_context ()
GList *	ags_audio_get_recall_container ()
void	ags_audio_set_recall_container ()
void	ags_audio_add_recall_container ()
void	ags_audio_remove_recall_container ()
GList *	ags_audio_get_play ()
void	ags_audio_set_play ()
GList *	ags_audio_get_recall ()
void	ags_audio_set_recall ()
void	ags_audio_add_recall ()
void	ags_audio_insert_recall ()
void	ags_audio_remove_recall ()
void	ags_audio_duplicate_recall ()
void	ags_audio_resolve_recall ()
void	ags_audio_init_recall ()
void	ags_audio_play_recall ()
void	ags_audio_done_recall ()
void	ags_audio_cancel_recall ()
void	ags_audio_cleanup_recall ()
GList *	ags_audio_start ()
void	ags_audio_stop ()
GList *	ags_audio_check_scope ()
GList *	ags_audio_collect_all_audio_ports ()
GList *	ags_audio_collect_all_audio_ports_by_specifier_and_context ()
void	ags_audio_open_audio_file_as_channel ()
void	ags_audio_open_audio_file_as_wave ()
void	ags_audio_open_midi_file_as_midi ()
void	ags_audio_open_midi_file_as_notation ()
void	ags_audio_recursive_set_property ()
void	ags_audio_recursive_run_stage ()
AgsAudio *	ags_audio_new ()
#define	AGS_AUDIO()
#define	AGS_AUDIO_CLASS()
#define	AGS_AUDIO_GET_CLASS()
#define	AGS_IS_AUDIO()
#define	AGS_IS_AUDIO_CLASS()
GType	ags_audio_flags_get_type ()
GType	ags_audio_get_type ()

Properties

int	absolute-key	Read / Write
guint	audio-channels	Read / Write
guint	audio-end-mapping	Read / Write
char *	audio-name	Read / Write
guint	audio-start-mapping	Read / Write
gpointer	automation	Read / Write
double	bpm	Read / Write
guint	buffer-size	Read / Write
gpointer	cursor	Read / Write
guint	denominator	Read / Write

guint	format	Read / Write
AgsInput *	input	Read
GObject *	input-audio-file	Read / Write
guint	input-lines	Read
GObject *	input-midi-file	Read / Write
guint	input-pads	Read / Write
GObject *	input-sequencer	Read / Write
GObject *	input-soundcard	Read / Write
gboolean	is-minor	Read / Write
guint	key	Read / Write
guint64	loop-end	Read / Write
guint64	loop-start	Read / Write
guint	max-audio-channels	Read / Write
guint	max-input-pads	Read / Write
guint	max-output-pads	Read / Write
gpointer	midi	Read / Write
guint	midi-channel	Read / Write
guint	midi-end-mapping	Read / Write
guint	midi-start-mapping	Read / Write
guint	min-audio-channels	Read / Write
guint	min-input-pads	Read / Write
guint	min-output-pads	Read / Write
gpointer	notation	Read / Write
guint	numerator	Read / Write
int	octave	Read / Write
guint64	offset	Read / Write
AgsOutput *	output	Read
GObject *	output-audio-file	Read / Write
guint	output-lines	Read
GObject *	output-midi-file	Read / Write
guint	output-pads	Read / Write
GObject *	output-sequencer	Read / Write
GObject *	output-soundcard	Read / Write
gpointer	play	Read / Write
AgsPlaybackDomain *	playback-domain	Read / Write
gpointer	preset	Read / Write
gpointer	recall	Read / Write
gpointer	recall-container	Read / Write
gpointer	recall-id	Read / Write
gpointer	recycling-context	Read / Write
guint	samplerate	Read / Write
gpointer	sf2-synth-generator	Read / Write
gpointer	sfz-synth-generator	Read / Write
guint	sharp-flats	Read / Write
gpointer	synth-generator	Read / Write
char *	time-signature	Read
gpointer	wave	Read / Write

Signals

void	cancel-recall	Run Last
gpointer	check-scope	Run Last
void	cleanup-recall	Run Last
void	done-recall	Run Last
void	duplicate-recall	Run Last

void	init-recall	Run Last
void	play-recall	Run Last
void	recursive-run-stage	Run Last
void	resolve-recall	Run Last
void	set-audio-channels	Run Last
void	set-pads	Run Last
gpointer	start	Run Last
void	stop	Run Last

Types and Values

enum	AgsAudioFlags
#define	AGS_TYPE_AUDIO
struct	AgsAudio
struct	AgsAudioClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsAudioFlags
GObject
&#x2570;&#x2500;&#x2500; AgsAudio
```

Implemented Interfaces

AgsAudio implements AgsConnectable.

Includes

```
#include <ags/audio/ags_audio.h>
```

Description

AgsAudio organizes AgsChannel objects either as input or output and is responsible of their alignment. The class can contain AgsRecall objects in order to perform computation on all channels or in audio context. Therefor exists AgsRecyclingContext acting as tree context.

At least one AgsRecallID is assigned to it and has one more if AGS_AUDIO_OUTPUT_HAS_RECYCLING is set as flag.

It hosts common storage objects like AgsNotation, AgsAutomation and AgsWave. In order to make use of AgsNotation or AgsWave you have to set AGS_AUDIO_OUTPUT_HAS_RECYCLING.

Make sure you have set the matching AgsSoundAbilityFlags flags using ags_audio_set_ability_flags(). This is going to setup threads per matching AgsSoundScope scope. See AgsPlaybackDomain and AgsPlayback.

Threads are usually run per AgsAudio with AGS_AUDIO_OUTPUT_HAS_RECYCLING. This are instruments, samplers, synthesizer, step sequencers or wave players. In other words the producer of your AgsAudioSignal.

The dispatcher function ags_audio_recursive_run_stage() is invoked as per AgsSoundScope threads, hosted by AgsAudio. The staging program can be modified. It calls then ags_channel_recursive_run_stage() per output channel.

AgsSoundStagingFlags is used to build your staging program for AgsAudioThread. The audio threads are created per AgsSoundScope and are provided by AgsAudio `playback-domain` property.

You have in general one AgsPlaybackDomain referring one AgsAudio and one AgsPlayback referring one AgsChannel.

AGS_AUDIO_SYNC and AGS_AUDIO_ASYNC AgsAudioFlags affect channel alignment. There 3 possible combinations available.

AGS_AUDIO_SYNC means channels go straight from input line to matching output line.

AGS_AUDIO_ASYNC means input channels are bundled in output i.e. many inputs of the same audio channel arrive at one output channel of the same audio channel.

AGS_AUDIO_SYNC and AGS_AUDIO_ASYNC combined is only available if AGS_AUDIO_OUTPUT_HAS_RECYCLING is set. It is actually the same behaviour as AGS_AUDIO_ASYNC except you can have more than one output pad.

`output-soundcard` and `input-soundcard` properties are AgsSoundcard implementations to read or write to a soundcard backend.

`output-sequencer` and `input-sequencer` properties are AgsSequencer implementations to read or write to a sequencer backend.

You may add AgsNotation objects using ags_audio_add_notation(). Make sure the AgsNotation timestamp offset occurs only once per audio channel and is correct padded using AGS_NOTATION_DEFAULT_OFFSET.

You may add AgsWave objects using ags_audio_add_wave(). Make sure the AgsWave timestamp offset occurs only once per line and is correct padded using relative offset. The relative offset is calculated as following:

relative_offset = AGS_WAVE_DEFAULT_BUFFER_LENGTH * samplerate;

You may add AgsAutomation objects using ags_audio_add_automation(). Make sure AgsAutomation timestamp offset occurs only once per channel type and line. If the context of the automation is AgsAudio `channel-type` is G_TYPE_NONE and `line` is 0.

To modify the alignment of output or input channels use ags_audio_set_audio_channels() and to increase pads ags_audio_set_pads().

There is a playback and recall context for hosting AgsRecall implementations. Usually you instantiate a recall for both contices. See `play` and `recall` properties.

Functions

AGS_AUDIO_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_GET_OBJ_MUTEX(obj) (&(((AgsAudio *) obj)->obj_mutex))
```

AGS_AUDIO_GET_PLAY_MUTEX()

```
#define AGS_AUDIO_GET_PLAY_MUTEX(obj) (&(((AgsAudio *) obj)->play_mutex))
```

AGS_AUDIO_GET_RECALL_MUTEX()

```
#define AGS_AUDIO_GET_RECALL_MUTEX(obj) (&(((AgsAudio *) obj)->recall_mutex))
```

ags_audio_get_obj_mutex ()

```
GRecMutex~*
ags_audio_get_obj_mutex (AgsAudio *audio);
```

Get object mutex.

Parameters

audio	the AgsAudio

Returns

the GRecMutex to lock *audio*

Since: 3.1.0

ags_audio_get_play_mutex ()

```
GRecMutex~*
ags_audio_get_play_mutex (AgsAudio *audio);
```

Get play mutex.

Parameters

audio	the AgsAudio

Returns

the GRecMutex to lock *audio* 's play property

Since: 3.1.0

ags_audio_get_recall_mutex ()

```
GRecMutex~*
ags_audio_get_recall_mutex (AgsAudio *audio);
```

Get recall mutex.

Parameters

audio	the AgsAudio

Returns

the GRecMutex to lock *audio* 's recall property

Since: 3.1.0

ags_audio_test_flags ()

```
gboolean
ags_audio_test_flags (AgsAudio *audio,
                      guint flags);
```

Test *flags* to be set on *audio* .

Parameters

audio	the AgsAudio	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_set_flags ()

```
void
ags_audio_set_flags (AgsAudio *audio,
                     guint flags);
```

Enable a feature of AgsAudio.

Parameters

audio	the AgsAudio	
flags	see enum AgsAudioFlags	

Since: 3.0.0

ags_audio_unset_flags ()

```
void
ags_audio_unset_flags (AgsAudio *audio,
                       guint flags);
```

Disable a feature of AgsAudio.

Parameters

audio	the AgsAudio	
flags	see enum AgsAudioFlags	

Since: 3.0.0

ags_audio_test_ability_flags ()

```
gboolean
ags_audio_test_ability_flags (AgsAudio *audio,
                              guint ability_flags);
```

Test *ability_flags* to be set on *audio* .

Parameters

audio	the AgsAudio	
ability_flags	the ability flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_set_ability_flags ()

```
void
ags_audio_set_ability_flags (AgsAudio *audio,
                             guint ability_flags);
```

Enable an ability of AgsAudio.

Parameters

audio	the AgsAudio	
ability_flags	see enum AgsSoundAbilityFlags	

Since: 3.0.0

ags_audio_unset_ability_flags ()

```
void
ags_audio_unset_ability_flags (AgsAudio *audio,
                               guint ability_flags);
```

Disable an ability of AgsAudio.

Parameters

audio	the AgsAudio	
ability_flags	see enum AgsSoundAbilityFlags	

Since: 3.0.0

ags_audio_test_behaviour_flags ()

```
gboolean
ags_audio_test_behaviour_flags (AgsAudio *audio,
                                guint behaviour_flags);
```

Test *behaviour_flags* to be set on *audio* .

Parameters

audio	the AgsAudio	
behaviour_flags	the behaviour flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_set_behaviour_flags ()

```
void
ags_audio_set_behaviour_flags (AgsAudio *audio,
                               guint behaviour_flags);
```

Set behaviour flags.

Parameters

audio	the AgsAudio	
behaviour_flags	the behaviour flags	

Since: 3.0.0

ags_audio_unset_behaviour_flags ()

```
void
ags_audio_unset_behaviour_flags (AgsAudio *audio,
                                 guint behaviour_flags);
```

Unset behaviour flags.

Parameters

audio	the AgsAudio	
behaviour_flags	the behaviour flags	

Since: 3.0.0

ags_audio_test_staging_flags ()

```
gboolean
ags_audio_test_staging_flags (AgsAudio *audio,
                              gint sound_scope,
                              guint staging_flags);
```

Test *staging_flags* to be set on *audio* .

Parameters

audio	the AgsAudio	
sound_scope	the AgsSoundScope to test	
staging_flags	the staging flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_set_staging_flags ()

```
void
ags_audio_set_staging_flags (AgsAudio *audio,
                             gint sound_scope,
                             guint staging_flags);
```

Set staging flags.

Parameters

audio	the AgsAudio	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	
staging_flags	the staging flags	

Since: 3.0.0

ags_audio_unset_staging_flags ()

```
void
ags_audio_unset_staging_flags (AgsAudio *audio,
                               gint sound_scope,
                               guint staging_flags);
```

Unset staging flags.

Parameters

audio	the AgsAudio	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	
staging_flags	the staging flags	

Since: 3.0.0

ags_audio_test_staging_completed ()

```
gboolean
ags_audio_test_staging_completed (AgsAudio *audio,
                                  gint sound_scope);
```

Test *sound_scope* to be completed on *audio* .

Parameters

audio	the AgsAudio	
sound_scope	the AgsSoundScope to test	

Returns

TRUE if completed, otherwise FALSE

Since: 3.3.0

ags_audio_set_staging_completed ()

```
void
ags_audio_set_staging_completed (AgsAudio *audio,
                                 gint sound_scope);
```

Set *sound_scope* to be completed.

Parameters

audio	the AgsAudio	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	

Since: 3.3.0

ags_audio_unset_staging_completed ()

```
void
ags_audio_unset_staging_completed (AgsAudio *audio,
                                   gint sound_scope);
```

Unset *sound_scope* to be completed.

Parameters

audio	the AgsAudio	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	

Since: 3.3.0

ags_audio_get_audio_name ()

```
gchar~*
ags_audio_get_audio_name (AgsAudio *audio);
```

Get audio name of *audio* .

Parameters

audio	the AgsAudio	

Returns

the audio name

Since: 3.1.0

ags_audio_set_audio_name ()

```
void
ags_audio_set_audio_name (AgsAudio *audio,
                          gchar *audio_name);
```

Set audio name of *audio* .

Parameters

audio	the AgsAudio	
audio_name	the audio name	

Since: 3.1.0

ags_audio_find_name ()

```
GList~*
ags_audio_find_name (GList *audio,
                     gchar *audio_name);
```

Find *audio_name* in *audio* .

Parameters

audio	the GList containing AgsAudio.	*[element-type AgsAudio.Audio][transfer none]*
audio_name	the audio name to find	

Returns

the next matching GList containing AgsAudio.

[element-type AgsAudio.Audio][transfer none]

Since: 3.0.0

ags_audio_get_max_audio_channels ()

```
guint
ags_audio_get_max_audio_channels (AgsAudio *audio);
```

Get maximum audio channels.

Parameters

audio	the AgsAudio	

Returns

the maximum audio channels

Since: 3.1.0

ags_audio_set_max_audio_channels ()

```
void
ags_audio_set_max_audio_channels (AgsAudio *audio,
                                  guint max_audio_channels);
```

Set maximum audio channels.

Parameters

audio	the AgsAudio	
max_audio_channels	maximum of audio channels	

Since: 3.0.0

ags_audio_get_max_output_pads ()

```
guint
ags_audio_get_max_output_pads (AgsAudio *audio);
```

Get maximum output pads.

Parameters

audio	the AgsAudio	

Returns

the maximum output pads

Since: 3.1.0

ags_audio_set_max_output_pads ()

```
void
ags_audio_set_max_output_pads (AgsAudio *audio,
                               guint max_output_pads);
```

Set maximum output pads.

Parameters

audio	the AgsAudio	
max_output_pads	maximum of output pads	

Since: 3.1.0

ags_audio_get_max_input_pads ()

```
guint
ags_audio_get_max_input_pads (AgsAudio *audio);
```

Get maximum input pads.

Parameters

audio	the AgsAudio	

Returns

the maximum input pads

Since: 3.1.0

ags_audio_set_max_input_pads ()

```
void
ags_audio_set_max_input_pads (AgsAudio *audio,
                              guint max_input_pads);
```

Set maximum input pads.

Parameters

audio	the AgsAudio	
max_input_pads	maximum of input pads	

Since: 3.1.0

ags_audio_set_max_pads ()

```
void
ags_audio_set_max_pads (AgsAudio *audio,
                        GType channel_type,
                        guint max_pads);
```

Set maximum pads of *channel_type* .

Parameters

audio	the AgsAudio	
channel_type	the GType of channel, either AGS_TYPE_OUTPUT or AGS_TYPE_INPUT	
max_pads	maximum of pads	

Since: 3.0.0

ags_audio_get_audio_channels ()

```
guint
ags_audio_get_audio_channels (AgsAudio *audio);
```

Get audio channels.

Parameters

audio	the AgsAudio	

Returns

the audio channels

Since: 3.1.0

ags_audio_set_audio_channels ()

```
void
ags_audio_set_audio_channels (AgsAudio *audio,
                              guint audio_channels,
                              guint audio_channels_old);
```

Resize audio channels AgsInput will be allocated first.

Parameters

audio	the AgsAudio	
audio_channels	new audio channels	
audio_channels_old	old audio channels	

Since: 3.0.0

ags_audio_get_output_pads ()

```
guint
ags_audio_get_output_pads (AgsAudio *audio);
```

Get output pads.

Parameters

audio	the AgsAudio	

Returns

the output pads

Since: 3.1.0

ags_audio_set_output_pads ()

```
void
ags_audio_set_output_pads (AgsAudio *audio,
                           guint output_pads);
```

Set output pads.

Parameters

audio	the AgsAudio	
output_pads	output pads	

Since: 3.1.0

ags_audio_get_input_pads ()

```
guint
ags_audio_get_input_pads (AgsAudio *audio);
```

Get input pads.

Parameters

audio	the AgsAudio	

Returns

the input pads

Since: 3.1.0

ags_audio_set_input_pads ()

```
void
ags_audio_set_input_pads (AgsAudio *audio,
                          guint input_pads);
```

Set input pads.

Parameters

audio	the AgsAudio	
input_pads	input pads	

Since: 3.1.0

ags_audio_set_pads ()

```
void
ags_audio_set_pads (AgsAudio *audio,
                    GType channel_type,
                    guint pads,
                    guint pads_old);
```

Set pad count for the apropriate *channel_type*

Parameters

audio	the AgsAudio	
channel_type	AGS_TYPE_INPUT or AGS_TYPE_OUTPUT	
pads	new pad count	
pads_old	old pad count	

Since: 3.0.0

ags_audio_get_output_lines ()

```
guint
ags_audio_get_output_lines (AgsAudio *audio);
```

Get output lines.

Parameters

audio	the AgsAudio	

Returns

the output lines

Since: 3.1.0

ags_audio_get_input_lines ()

```
guint
ags_audio_get_input_lines (AgsAudio *audio);
```

Get input lines.

Parameters

audio	the AgsAudio

Returns

the input lines

Since: 3.1.0

ags_audio_get_output ()

```
AgsChannel~*
ags_audio_get_output (AgsAudio *audio);
```

Get the output object of *audio* .

Parameters

audio	the AgsAudio

Returns

the output.

[transfer full]

Since: 3.1.0

ags_audio_get_input ()

```
AgsChannel~*
ags_audio_get_input (AgsAudio *audio);
```

Get the input object of *audio* .

Parameters

audio	the AgsAudio

Returns

the input.

[transfer full]

Since: 3.1.0

ags_audio_get_output_soundcard ()

```
GObject~*
ags_audio_get_output_soundcard (AgsAudio *audio);
```

Get the output soundcard object of *audio* .

Parameters

audio	the AgsAudio	

Returns

the output soundcard.

[transfer full]

Since: 3.1.0

ags_audio_set_output_soundcard ()

```
void
ags_audio_set_output_soundcard (AgsAudio *audio,
                                GObject *output_soundcard);
```

Set the output soundcard object of *audio* .

Parameters

audio	the AgsAudio	
output_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_audio_get_input_soundcard ()

```
GObject~*
ags_audio_get_input_soundcard (AgsAudio *audio);
```

Get the input soundcard object of *audio* .

Parameters

audio	the AgsAudio	

Returns

the input soundcard.

[transfer full]

Since: 3.1.0

ags_audio_set_input_soundcard ()

```
void
ags_audio_set_input_soundcard (AgsAudio *audio,
                               GObject *input_soundcard);
```

Set the input soundcard object of *audio* .

Parameters

audio	an AgsAudio	
input_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_audio_get_output_sequencer ()

```
GObject~*
ags_audio_get_output_sequencer (AgsAudio *audio);
```

Get the output sequencer object of *audio* .

Parameters

audio	the AgsAudio	

Returns

the output sequencer.

[transfer full]

Since: 3.1.0

ags_audio_set_output_sequencer ()

```
void
ags_audio_set_output_sequencer (AgsAudio *audio,
                                GObject *output_sequencer);
```

Set the output sequencer object of *audio* .

Parameters

audio	an AgsAudio	
output_sequencer	the GObject implementing AgsSequencer	

Since: 3.0.0

ags_audio_get_input_sequencer ()

```
GObject~*
ags_audio_get_input_sequencer (AgsAudio *audio);
```

Get the input sequencer object of *audio* .

Parameters

audio	the AgsAudio

Returns

the input sequencer.

[transfer full]

Since: 3.1.0

ags_audio_set_input_sequencer ()

```
void
ags_audio_set_input_sequencer (AgsAudio *audio,
                               GObject *input_sequencer);
```

Set the input sequencer object of `audio`.

Parameters

audio	an AgsAudio	
input_sequencer	the GObject implementing AgsSequencer	

Since: 3.0.0

ags_audio_get_samplerate ()

```
guint
ags_audio_get_samplerate (AgsAudio *audio);
```

Gets samplerate.

Parameters

audio	the AgsAudio

Returns

the samplerate

Since: 3.1.0

ags_audio_set_samplerate ()

```
void
ags_audio_set_samplerate (AgsAudio *audio,
                          guint samplerate);
```

Sets samplerate.

Parameters

audio	the AgsAudio	
samplerate	the samplerate	

Since: 3.0.0

ags_audio_get_buffer_size ()

```
guint
ags_audio_get_buffer_size (AgsAudio *audio);
```

Gets buffer size.

Parameters

audio	the AgsAudio	

Returns

the buffer size

Since: 3.1.0

ags_audio_set_buffer_size ()

```
void
ags_audio_set_buffer_size (AgsAudio *audio,
                           guint buffer_size);
```

Sets buffer length.

Parameters

audio	the AgsAudio	
buffer_size	the buffer length	

Since: 3.0.0

ags_audio_get_format ()

```
guint
ags_audio_get_format (AgsAudio *audio);
```

Gets format.

Parameters

audio	the AgsAudio	

Returns

the format

Since: 3.1.0

ags_audio_set_format ()

```
void
ags_audio_set_format (AgsAudio *audio,
                      guint format);
```

Sets buffer length.

Parameters

audio	the AgsAudio	
format	the format	

Since: 3.0.0

ags_audio_get_bpm ()

```
gdouble
ags_audio_get_bpm (AgsAudio *audio);
```

Gets bpm.

Parameters

audio	the AgsAudio	

Returns

the bpm

Since: 3.1.0

ags_audio_set_bpm ()

```
void
ags_audio_set_bpm (AgsAudio *audio,
                   gdouble bpm);
```

Sets bpm.

Parameters

audio	the AgsAudio	
bpm	the bpm	

Since: 3.1.0

ags_audio_get_audio_start_mapping ()

```
guint
ags_audio_get_audio_start_mapping (AgsAudio *audio);
```

Gets audio start mapping.

Parameters

audio	the AgsAudio

Returns

the audio start mapping

Since: 3.1.0

ags_audio_set_audio_start_mapping ()

```
void
ags_audio_set_audio_start_mapping (AgsAudio *audio,
                                   guint audio_start_mapping);
```

Sets audio start mapping.

Parameters

audio	the AgsAudio
audio_start_mapping	the audio start mapping

Since: 3.1.0

ags_audio_get_midi_start_mapping ()

```
guint
ags_audio_get_midi_start_mapping (AgsAudio *audio);
```

Gets midi start mapping.

Parameters

audio	the AgsAudio

Returns

the midi start mapping

Since: 3.1.0

ags_audio_set_midi_start_mapping ()

```
void
ags_audio_set_midi_start_mapping (AgsAudio *audio,
                                  guint midi_start_mapping);
```

Sets midi start mapping.

Parameters

audio	the AgsAudio	
midi_start_mapping	the midi start mapping	

Since: 3.1.0

ags_audio_get_midi_channel ()

```
guint
ags_audio_get_midi_channel (AgsAudio *audio);
```

Gets midi channel.

Parameters

audio	the AgsAudio	

Returns

the midi channel

Since: 3.1.0

ags_audio_set_midi_channel ()

```
void
ags_audio_set_midi_channel (AgsAudio *audio,
                            guint midi_channel);
```

Sets midi channel.

Parameters

audio	the AgsAudio	
midi_channel	the midi channel	

Since: 3.1.0

ags_audio_get_numerator ()

```
guint
ags_audio_get_numerator (AgsAudio *audio);
```

Gets numerator.

Parameters

audio	the AgsAudio

Returns

the numerator

Since: 3.1.0

ags_audio_set_numerator ()

```
void
ags_audio_set_numerator (AgsAudio *audio,
                         guint numerator);
```

Sets numerator.

Parameters

audio	the AgsAudio
numerator	the numerator

Since: 3.1.0

ags_audio_get_denominator ()

```
guint
ags_audio_get_denominator (AgsAudio *audio);
```

Gets denominator.

Parameters

audio	the AgsAudio

Returns

the denominator

Since: 3.1.0

ags_audio_set_denominator ()

```
void
ags_audio_set_denominator (AgsAudio *audio,
                           guint denominator);
```

Sets denominator.

Parameters

audio	the AgsAudio	
denominator	the denominator	

Since: 3.1.0

ags_audio_get_time_signature ()

```
gchar~*
ags_audio_get_time_signature (AgsAudio *audio);
```

Gets time signature.

Parameters

audio	the AgsAudio	

Returns

the time signature

Since: 3.1.0

ags_audio_set_time_signature ()

```
void
ags_audio_set_time_signature (AgsAudio *audio,
                              gchar *time_signature);
```

Sets time signature.

Parameters

audio	the AgsAudio	
time_signature	the time signature	

Since: 3.1.0

ags_audio_get_is_minor ()

```
gboolean
ags_audio_get_is_minor (AgsAudio *audio);
```

Gets is minor.

Parameters

audio	the AgsAudio	

Returns

is minor

Since: 3.1.0

ags_audio_set_is_minor ()

```
void
ags_audio_set_is_minor (AgsAudio *audio,
                        gboolean is_minor);
```

Sets is minor.

Parameters

audio	the AgsAudio	
is_minor	is minor	

Since: 3.1.0

ags_audio_get_sharp_flats ()

```
guint
ags_audio_get_sharp_flats (AgsAudio *audio);
```

Gets sharp flats.

Parameters

audio	the AgsAudio	

Returns

the sharp flats

Since: 3.1.0

ags_audio_set_sharp_flats ()

```
void
ags_audio_set_sharp_flats (AgsAudio *audio,
                           guint sharp_flats);
```

Sets sharp flats.

Parameters

audio	the AgsAudio	
sharp_flats	the sharp flats	

Since: 3.1.0

ags_audio_get_octave ()

```
gint
ags_audio_get_octave (AgsAudio *audio);
```

Gets octave.

Parameters

audio	the AgsAudio

Returns

the octave

Since: 3.1.0

ags_audio_set_octave ()

```
void
ags_audio_set_octave (AgsAudio *audio,
                      gint octave);
```

Sets octave.

Parameters

audio	the AgsAudio
octave	the octave

Since: 3.1.0

ags_audio_get_key ()

```
guint
ags_audio_get_key (AgsAudio *audio);
```

Gets key.

Parameters

audio	the AgsAudio

Returns

the key

Since: 3.1.0

ags_audio_set_key ()

```
void
ags_audio_set_key (AgsAudio *audio,
                   guint key);
```

Sets key.

Parameters

audio	the AgsAudio	
key	the key	

Since: 3.1.0

ags_audio_get_absolute_key ()

```
gint
ags_audio_get_absolute_key (AgsAudio *audio);
```

Gets absolute key.

Parameters

audio	the AgsAudio	

Returns

the absolute key

Since: 3.1.0

ags_audio_set_absolute_key ()

```
void
ags_audio_set_absolute_key (AgsAudio *audio,
                            gint absolute_key);
```

Sets absolute key.

Parameters

audio	the AgsAudio	
absolute_key	the absolute key	

Since: 3.1.0

ags_audio_get_preset ()

```
GList~*
ags_audio_get_preset (AgsAudio *audio);
```

Get preset.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsPreset.

[element-type AgsAudio.Preset][transfer full]

Since: 3.1.0

ags_audio_set_preset ()

```
void
ags_audio_set_preset (AgsAudio *audio,
                      GList *preset);
```

Set preset by replacing existing.

Parameters

audio	the AgsAudio	
preset	the GList containing AgsPreset.	*[element-type AgsAudio.Preset][transfer full]*

Since: 3.1.0

ags_audio_add_preset ()

```
void
ags_audio_add_preset (AgsAudio *audio,
                      GObject *preset);
```

Adds a preset.

Parameters

audio	the AgsAudio	
preset	an AgsPreset	

Since: 3.0.0

ags_audio_remove_preset ()

```
void
ags_audio_remove_preset (AgsAudio *audio,
                         GObject *preset);
```

Removes a preset.

Parameters

audio	the AgsAudio	
preset	an AgsPreset	

Since: 3.0.0

ags_audio_get_playback_domain ()

```
GObject~*
ags_audio_get_playback_domain (AgsAudio *audio);
```

Get playback domain.

Parameters

audio	the AgsAudio	

Returns

the AgsPlaybackDomain.

[transfer full]

Since: 3.1.0

ags_audio_set_playback_domain ()

```
void
ags_audio_set_playback_domain (AgsAudio *audio,
                               GObject *playback_domain);
```

Set playback domain.

Parameters

audio	the AgsAudio	
playback_domain	the AgsPlaybackDomain	

Since: 3.1.0

ags_audio_get_synth_generator ()

```
GList~*
ags_audio_get_synth_generator (AgsAudio *audio);
```

Get synth generator.

Parameters

audio	the AgsAudio	

Returns

the GList containing AgsSynthGenerator.

[element-type AgsAudio.SynthGenerator][transfer full]

Since: 3.1.0

ags_audio_set_synth_generator ()

```
void
ags_audio_set_synth_generator (AgsAudio *audio,
                               GList *synth_generator);
```

Set synth generator by replacing existing.

Parameters

audio	the AgsAudio	
synth_generator	the GList containing AgsSynthGenerator.	*[element-type AgsAudio.SynthGenerator][transfer full]*

Since: 3.1.0

ags_audio_add_synth_generator ()

```
void
ags_audio_add_synth_generator (AgsAudio *audio,
                               GObject *synth_generator);
```

Adds a synth generator.

Parameters

audio	the AgsAudio	
synth_generator	an AgsSynthGenerator	

Since: 3.0.0

ags_audio_remove_synth_generator ()

```
void
ags_audio_remove_synth_generator (AgsAudio *audio,
                                  GObject *synth_generator);
```

Removes a synth generator.

Parameters

audio	the AgsAudio	
synth_generator	an AgsSynthGenerator	

Since: 3.0.0

ags_audio_get_sf2_synth_generator ()

```
GList~*
ags_audio_get_sf2_synth_generator (AgsAudio *audio);
```

Get SF2 synth generator.

Parameters

audio	the AgsAudio	

Returns

the GList containing AgsSF2SynthGenerator.

[element-type AgsAudio.SF2SynthGenerator][transfer full]

Since: 3.4.0

ags_audio_set_sf2_synth_generator ()

```
void
ags_audio_set_sf2_synth_generator (AgsAudio *audio,
                                   GList *sf2_synth_generator);
```

Set SF2 synth generator by replacing existing.

Parameters

audio	the AgsAudio	
sf2_synth_generator	the GList containing AgsSF2SynthGenerator.	*[element-type AgsAudio.SF2SynthGenerator][transfer full]*

Since: 3.4.0

ags_audio_add_sf2_synth_generator ()

```
void
ags_audio_add_sf2_synth_generator (AgsAudio *audio,
                                   GObject *sf2_synth_generator);
```

Adds a SF2 synth generator.

Parameters

audio	the AgsAudio	
sf2_synth_generator	an AgsSF2SynthGenerator	

Since: 3.4.0

ags_audio_remove_sf2_synth_generator ()

```
void
ags_audio_remove_sf2_synth_generator (AgsAudio *audio,
                                      GObject *sf2_synth_generator);
```

Removes a SF2 synth generator.

Parameters

audio	the AgsAudio	
sf2_synth_generator	an AgsSF2SynthGenerator	

Since: 3.4.0

ags_audio_get_sfz_synth_generator ()

```
GList~*
ags_audio_get_sfz_synth_generator (AgsAudio *audio);
```

Get SFZ synth generator.

Parameters

audio	the AgsAudio	

Returns

the GList containing AgsSFZSynthGenerator.

[element-type AgsAudio.SFZSynthGenerator][transfer full]

Since: 3.4.0

ags_audio_set_sfz_synth_generator ()

```
void
ags_audio_set_sfz_synth_generator (AgsAudio *audio,
                                   GList *sfz_synth_generator);
```

Set SFZ synth generator by replacing existing.

Parameters

audio	the AgsAudio	
sfz_synth_generator	the GList containing AgsSFZSynthGenerator.	*[element-type AgsAudio.SFZSynthGenerator][transfer full]*

Since: 3.4.0

ags_audio_add_sfz_synth_generator ()

```
void
ags_audio_add_sfz_synth_generator (AgsAudio *audio,
                                   GObject *sfz_synth_generator);
```

Adds a SFZ synth generator.

Parameters

audio	the AgsAudio	
sfz_synth_generator	an AgsSFZSynthGenerator	

Since: 3.4.0

ags_audio_remove_sfz_synth_generator ()

```
void
ags_audio_remove_sfz_synth_generator (AgsAudio *audio,
                                      GObject *sfz_synth_generator);
```

Removes a SFZ synth generator.

Parameters

audio	the AgsAudio	
sfz_synth_generator	an AgsSFZSynthGenerator	

Since: 3.4.0

ags_audio_get_cursor ()

```
GList~*
ags_audio_get_cursor (AgsAudio *audio);
```

Get cursor.

Parameters

audio	the AgsAudio	

Returns

the GList containig GObject implementing AgsCursor.

[element-type GObject][transfer full]

Since: 3.1.0

ags_audio_set_cursor ()

```
void
ags_audio_set_cursor (AgsAudio *audio,
                      GList *cursor);
```

Set cursor by replacing existing.

Parameters

audio	the AgsAudio	
cursor	the GList containing AgsCursor.	*[element-type GObject][transfer full]*

Since: 3.1.0

ags_audio_add_cursor ()

```
void
ags_audio_add_cursor (AgsAudio *audio,
                      GObject *cursor);
```

Adds a cursor.

Parameters

audio	the AgsAudio	
cursor	the GObject implementing AgsCursor	

Since: 3.0.0

ags_audio_remove_cursor ()

```
void
ags_audio_remove_cursor (AgsAudio *audio,
                         GObject *cursor);
```

Removes a cursor.

Parameters

audio	the AgsAudio	
cursor	the GObject implementing AgsCursor	

Since: 3.0.0

ags_audio_get_notation ()

```
GList~*
ags_audio_get_notation (AgsAudio *audio);
```

Get notation.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsNotation.

[element-type AgsAudio.Notation][transfer full]

Since: 3.1.0

ags_audio_set_notation ()

```
void
ags_audio_set_notation (AgsAudio *audio,
                        GList *notation);
```

Set notation by replacing existing.

Parameters

audio	the AgsAudio	
notation	the GList containing AgsNotation.	*[element-type AgsAudio.Notation][transfer full]*

Since: 3.1.0

ags_audio_add_notation ()

```
void
ags_audio_add_notation (AgsAudio *audio,
                        GObject *notation);
```

Adds a notation.

Parameters

audio	the AgsAudio	
notation	the AgsNotation	

Since: 3.0.0

ags_audio_remove_notation ()

```
void
ags_audio_remove_notation (AgsAudio *audio,
                           GObject *notation);
```

Removes a notation.

Parameters

audio	the AgsAudio	
notation	the AgsNotation	

Since: 3.0.0

ags_audio_get_automation_port ()

```
gchar~**
ags_audio_get_automation_port (AgsAudio *audio);
```

Get automation port.

Parameters

audio	the AgsAudio	

Returns

the string vector containing automation ports.

[transfer full]

Since: 3.1.0

ags_audio_set_automation_port ()

```
void
ags_audio_set_automation_port (AgsAudio *audio,
                               gchar **automation_port);
```

Get automation port.

Parameters

audio	the AgsAudio	
automation_port	the string vector containing automation ports.	*[transfer full]*

Since: 3.1.0

ags_audio_add_automation_port ()

```
void
ags_audio_add_automation_port (AgsAudio *audio,
                               gchar *control_name);
```

Adds an automation port.

Parameters

audio	the AgsAudio	
control_name	the control name	

Since: 3.0.0

ags_audio_remove_automation_port ()

```
void
ags_audio_remove_automation_port (AgsAudio *audio,
                                  gchar *control_name);
```

Removes an automation port.

Parameters

audio	the AgsAudio	
control_name	the control name	

Since: 3.0.0

ags_audio_get_automation ()

```
GList~*
ags_audio_get_automation (AgsAudio *audio);
```

Get automation.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsAutomation.

[element-type AgsAudio.Automation][transfer full]

Since: 3.1.0

ags_audio_set_automation ()

```
void
ags_audio_set_automation (AgsAudio *audio,
                          GList *automation);
```

Set automation by replacing existing.

Parameters

audio	the AgsAudio	
automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer full]*

Since: 3.1.0

ags_audio_add_automation ()

```
void
ags_audio_add_automation (AgsAudio *audio,
                          GObject *automation);
```

Adds an automation.

Parameters

audio	the AgsAudio	
automation	the AgsAutomation	

Since: 3.0.0

ags_audio_remove_automation ()

```
void
ags_audio_remove_automation (AgsAudio *audio,
                             GObject *automation);
```

Removes an automation.

Parameters

audio	the AgsAudio	
automation	the AgsAutomation	

Since: 3.0.0

ags_audio_get_wave ()

```
GList~*
ags_audio_get_wave (AgsAudio *audio);
```

Get wave.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsWave.

[element-type AgsAudio.Wave][transfer full]

Since: 3.1.0

ags_audio_set_wave ()

```
void
ags_audio_set_wave (AgsAudio *audio,
                    GList *wave);
```

Set wave by replacing existing.

Parameters

audio	the AgsAudio	
wave	the GList containing AgsWave.	*[element-type AgsAudio.Wave][transfer full]*

Since: 3.1.0

ags_audio_add_wave ()

```
void
ags_audio_add_wave (AgsAudio *audio,
                    GObject *wave);
```

Adds a wave.

Parameters

audio	the AgsAudio	
wave	the AgsWave	

Since: 3.0.0

ags_audio_remove_wave ()

```
void
ags_audio_remove_wave (AgsAudio *audio,
                       GObject *wave);
```

Removes a wave.

Parameters

audio	the AgsAudio	
wave	the AgsWave	

Since: 3.0.0

ags_audio_get_output_audio_file ()

```
GObject~*
ags_audio_get_output_audio_file (AgsAudio *audio);
```

Get output audio file.

Parameters

audio	the AgsAudio	

Returns

the GObject.

[transfer full]

Since: 3.1.0

ags_audio_set_output_audio_file ()

```
void
ags_audio_set_output_audio_file (AgsAudio *audio,
                                 GObject *output_audio_file);
```

Set output audio file.

Parameters

audio	the AgsAudio	
output_audio_file	the GObject	

Since: 3.1.0

ags_audio_get_input_audio_file ()

```
GObject~*
ags_audio_get_input_audio_file (AgsAudio *audio);
```

Get input audio file.

Parameters

audio	the AgsAudio

Returns

the GObject.

[transfer full]

Since: 3.1.0

ags_audio_set_input_audio_file ()

```
void
ags_audio_set_input_audio_file (AgsAudio *audio,
                                GObject *input_audio_file);
```

Set input audio file.

Parameters

audio	the AgsAudio
input_audio_file	the GObject

Since: 3.1.0

ags_audio_get_instrument_name ()

```
gchar~*
ags_audio_get_instrument_name (AgsAudio *audio);
```

Gets instrument name.

Parameters

audio	the AgsAudio

Returns

the instrument name

Since: 3.1.0

ags_audio_set_instrument_name ()

```
void
ags_audio_set_instrument_name (AgsAudio *audio,
                               gchar *instrument_name);
```

Sets instrument name.

Parameters

audio	the AgsAudio	
instrument_name	the instrument name	

Since: 3.1.0

ags_audio_get_track_name ()

```
gchar~*
ags_audio_get_track_name (AgsAudio *audio);
```

Gets track name.

Parameters

audio	the AgsAudio	

Returns

the track name

Since: 3.1.0

ags_audio_set_track_name ()

```
void
ags_audio_set_track_name (AgsAudio *audio,
                          gchar *track_name);
```

Sets track name.

Parameters

audio	the AgsAudio	
track_name	the track name	

Since: 3.1.0

ags_audio_get_midi ()

```
GList~*
ags_audio_get_midi (AgsAudio *audio);
```

Get midi.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsMidi.

[element-type AgsAudio.Midi][transfer full]

Since: 3.1.0

ags_audio_set_midi ()

```
void
ags_audio_set_midi (AgsAudio *audio,
                    GList *midi);
```

Set midi by replacing existing.

Parameters

audio	the AgsAudio	
midi	the GList containing AgsMidi.	*[element-type AgsAudio.Midi][transfer full]*

Since: 3.1.0

ags_audio_add_midi ()

```
void
ags_audio_add_midi (AgsAudio *audio,
                    GObject *midi);
```

Adds a midi.

Parameters

audio	the AgsAudio	
midi	the AgsMidi	

Since: 3.0.0

ags_audio_remove_midi ()

```
void
ags_audio_remove_midi (AgsAudio *audio,
                       GObject *midi);
```

Removes a midi.

Parameters

audio	the AgsAudio	
midi	the AgsMidi	

Since: 3.0.0

ags_audio_get_output_midi_file ()

```
GObject~*
ags_audio_get_output_midi_file (AgsAudio *audio);
```

Get output MIDI file.

Parameters

audio	the AgsAudio	

Returns

the GObject.

[transfer full]

Since: 3.1.0

ags_audio_set_output_midi_file ()

```
void
ags_audio_set_output_midi_file (AgsAudio *audio,
                                GObject *output_midi_file);
```

Set output MIDI file.

Parameters

audio	the AgsAudio	
output_midi_file	the GObject	

Since: 3.1.0

ags_audio_get_input_midi_file ()

```
GObject~*
ags_audio_get_input_midi_file (AgsAudio *audio);
```

Get input MIDI file.

Parameters

audio	the AgsAudio	

Returns

the GObject.

[transfer full]

Since: 3.1.0

ags_audio_set_input_midi_file ()

```
void
ags_audio_set_input_midi_file (AgsAudio *audio,
                               GObject *input_midi_file);
```

Set input MIDI file.

Parameters

audio	the AgsAudio	
input_midi_file	the GObject	

Since: 3.1.0

ags_audio_get_recall_id ()

```
GList~*
ags_audio_get_recall_id (AgsAudio *audio);
```

Get recall id.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsRecallID.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.1.0

ags_audio_set_recall_id ()

```
void
ags_audio_set_recall_id (AgsAudio *audio,
                         GList *recall_id);
```

Set recall id by replacing existing.

Parameters

audio	the AgsAudio	
recall_id	the GList containing AgsRecallID.	*[element-type AgsAudio.RecallID][transfer full]*

Since: 3.1.0

ags_audio_add_recall_id ()

```
void
ags_audio_add_recall_id (AgsAudio *audio,
                         GObject *recall_id);
```

Adds a recall id.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_audio_remove_recall_id ()

```
void
ags_audio_remove_recall_id (AgsAudio *audio,
                            GObject *recall_id);
```

Removes a recall id.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_audio_get_recycling_context ()

```
GList~*
ags_audio_get_recycling_context (AgsAudio *audio);
```

Get recycling_context.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsRecyclingContext.

[element-type AgsAudio.RecyclingContext][transfer full]

Since: 3.1.0

ags_audio_set_recycling_context ()

```
void
ags_audio_set_recycling_context (AgsAudio *audio,
                                 GList *recycling_context);
```

Set recycling_context by replacing existing.

Parameters

audio	the AgsAudio	
recycling_context	the GList containing AgsRecyclingContext.	*[element-type AgsAudio.RecyclingContext][transfer full]*

Since: 3.1.0

ags_audio_add_recycling_context ()

```
void
ags_audio_add_recycling_context (AgsAudio *audio,
                                 GObject *recycling_context);
```

Adds a recycling context.

Parameters

audio	the AgsAudio	
recycling_context	the AgsRecyclingContext	

Since: 3.0.0

ags_audio_remove_recycling_context ()

```
void
ags_audio_remove_recycling_context (AgsAudio *audio,
                                    GObject *recycling_context);
```

Removes a recycling context.

Parameters

audio	the AgsAudio	
recycling_context	the AgsRecyclingContext	

Since: 3.0.0

ags_audio_get_recall_container ()

```
GList~*
ags_audio_get_recall_container (AgsAudio *audio);
```

Get recall_container.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsRecallContainer.

[element-type AgsAudio.RecallContainer][transfer full]

Since: 3.1.0

ags_audio_set_recall_container ()

```
void
ags_audio_set_recall_container (AgsAudio *audio,
                                GList *recall_container);
```

Set recall_container by replacing existing.

Parameters

audio	the AgsAudio	
recall_container	the GList containing AgsRecallContainer.	*[element-type AgsAudio.RecallContainer][transfer full]*

Since: 3.1.0

ags_audio_add_recall_container ()

```
void
ags_audio_add_recall_container (AgsAudio *audio,
                                GObject *recall_container);
```

Adds a recall container.

Parameters

audio	the AgsAudio	
recall_container	the AgsRecallContainer	

Since: 3.0.0

ags_audio_remove_recall_container ()

```
void
ags_audio_remove_recall_container (AgsAudio *audio,
                                   GObject *recall_container);
```

Removes a recall container.

Parameters

audio	the AgsAudio	
recall_container	the AgsRecallContainer	

Since: 3.0.0

ags_audio_get_play ()

```
GList~*
ags_audio_get_play (AgsAudio *audio);
```

Get play.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsRecall.

[element-type AgsAudio.Recall][transfer full]

Since: 3.1.0

ags_audio_set_play ()

```
void
ags_audio_set_play (AgsAudio *audio,
                    GList *play);
```

Set play by replacing existing.

Parameters

audio	the AgsAudio	
play	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer full]*

Since: 3.1.0

ags_audio_get_recall ()

```
GList~*
ags_audio_get_recall (AgsAudio *audio);
```

Get recall.

Parameters

audio	the AgsAudio	

Returns

the GList containig AgsRecall.

[element-type AgsAudio.Recall][transfer full]

Since: 3.1.0

ags_audio_set_recall ()

```
void
ags_audio_set_recall (AgsAudio *audio,
                      GList *recall);
```

Set recall by replacing existing.

Parameters

audio	the AgsAudio	
recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer full]*

Since: 3.1.0

ags_audio_add_recall ()

```
void
ags_audio_add_recall (AgsAudio *audio,
                      GObject *recall,
                      gboolean play_context);
```

Adds a recall to `audio`.

Parameters

audio	the AgsAudio	
recall	the AgsRecall	
play_context	if TRUE play context, else if FALSE recall context	

Since: 3.0.0

ags_audio_insert_recall ()

```
void
ags_audio_insert_recall (AgsAudio *audio,
                         GObject *recall,
                         gboolean play_context,
                         gint position);
```

Insert *recall* at *position* in *audio* 's *play_context* .

Parameters

audio	the AgsAudio	
recall	the AgsRecall	
play_context	if TRUE play context, else if FALSE recall context	
position	the position	

Since: 3.3.0

ags_audio_remove_recall ()

```
void
ags_audio_remove_recall (AgsAudio *audio,
                         GObject *recall,
                         gboolean play_context);
```

Removes a recall from *audio* .

Parameters

audio	the AgsAudio	
recall	the AgsRecall	
play_context	if TRUE play context, else if FALSE recall context	

Since: 3.0.0

ags_audio_duplicate_recall ()

```
void
ags_audio_duplicate_recall (AgsAudio *audio,
                            AgsRecallID *recall_id,
                            guint pad,
                            guint audio_channel,
                            guint line);
```

Duplicate all AgsRecall templates of *audio* .

Parameters

audio	the AgsAudio	
recall_id	an AgsRecallID	
pad	the pad	
audio_channel	the audio channel	
line	the line	

Since: 3.0.0

ags_audio_resolve_recall ()

```
void
ags_audio_resolve_recall (AgsAudio *audio,
                          AgsRecallID *recall_id);
```

Performs resolving of recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID to use	

Since: 3.0.0

ags_audio_init_recall ()

```
void
ags_audio_init_recall (AgsAudio *audio,
                       AgsRecallID *recall_id,
                       guint staging_flags);
```

Initializes the recalls of *audio*

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID to use or NULL	
staging_flags	staging flags, see AgsSoundStagingFlags	

Since: 3.0.0

ags_audio_play_recall ()

```
void
ags_audio_play_recall (AgsAudio *audio,
                       AgsRecallID *recall_id,
                       guint staging_flags);
```

Performs play for the specified *staging_flags* .

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID to apply to	
staging_flags	staging flags, see AgsSoundStagingFlags	

Since: 3.0.0

ags_audio_done_recall ()

```
void
ags_audio_done_recall (AgsAudio *audio,
                       AgsRecallID *recall_id);
```

Done processing audio data.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID to apply to	

Since: 3.0.0

ags_audio_cancel_recall ()

```
void
ags_audio_cancel_recall (AgsAudio *audio,
                         AgsRecallID *recall_id);
```

Cancel processing audio data.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID to apply to	

Since: 3.0.0

ags_audio_cleanup_recall ()

```
void
ags_audio_cleanup_recall (AgsAudio *audio,
                          AgsRecallID *recall_id);
```

Cleanup processing audio data.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID to apply to	

Since: 3.0.0

ags_audio_start ()

```
GList~*
ags_audio_start (AgsAudio *audio,
                 gint sound_scope);
```

Is emitted as audio is started.

Parameters

audio	the AgsAudio	
sound_scope	the scope	

Returns

the GList containing AgsRecallID.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.0.0

ags_audio_stop ()

```
void
ags_audio_stop (AgsAudio *audio,
                GList *recall_id,
                gint sound_scope);
```

Is emitted as playing audio is stopped.

Parameters

audio	the AgsAudio	
recall_id	the GList containing AgsRecallID.	*[element-type AgsAudio.RecallID][transfer none]*
sound_scope	the scope	

Since: 3.0.0

ags_audio_check_scope ()

```
GList~*
ags_audio_check_scope (AgsAudio *audio,
                       gint sound_scope);
```

Check scope's recall id.

Parameters

audio	the AgsAudio	
sound_scope	the scope	

Returns

the scope's recall id of `audio`.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.0.0

ags_audio_collect_all_audio_ports ()

```
GList~*
ags_audio_collect_all_audio_ports (AgsAudio *audio);
```

Retrieve all ports of AgsAudio.

Parameters

audio	the AgsAudio	

Returns

a new GList containing AgsPort.

[element-type AgsAudio.Port][transfer full]

Since: 3.0.0

ags_audio_collect_all_audio_ports_by_specifier_and_context ()

```
GList~*
ags_audio_collect_all_audio_ports_by_specifier_and_context
                                (AgsAudio *audio,
                                 gchar *specifier,
                                 gboolean play_context);
```

Retrieve specified port of AgsAudio

Parameters

audio	an AgsAudio	
specifier	the port's name	
play_context	either TRUE for play or FALSE for recall	

Returns

a GList of AgsPort if found, otherwise NULL.

[element-type AgsAudio.Port][transfer full]

Since: 3.0.0

ags_audio_open_audio_file_as_channel ()

```
void
ags_audio_open_audio_file_as_channel (AgsAudio *audio,
                                      GSList *filename,
                                      gboolean overwrite_channels,
                                      gboolean create_channels);
```

Open some files.

Parameters

audio	the AgsAudio	
filename	the files to open.	*[element-type utf8][transfer none]*
overwrite_channels	if existing channels should be assigned	
create_channels	if new channels should be created as not fitting if combined with `overwrite_channels`	

Since: 3.0.0

ags_audio_open_audio_file_as_wave ()

```
void
ags_audio_open_audio_file_as_wave (AgsAudio *audio,
                                   const gchar *filename,
                                   gboolean overwrite_channels,
                                   gboolean create_channels);
```

ags_audio_open_midi_file_as_midi ()

```
void
ags_audio_open_midi_file_as_midi (AgsAudio *audio,
                                  const gchar *filename,
                                  const gchar *instrument,
                                  const gchar *track_name,
                                  guint midi_channel);
```

ags_audio_open_midi_file_as_notation ()

```
void
ags_audio_open_midi_file_as_notation (AgsAudio *audio,
                                      const gchar *filename,
                                      const gchar *instrument,
                                      const gchar *track_name,
                                      guint midi_channel);
```

ags_audio_recursive_set_property ()

```
void
ags_audio_recursive_set_property (AgsAudio *audio,
                                  gint n_params,
                                  const gchar *parameter_name[],
                                  const GValue value[]);
```

Recursive set property for AgsAudio.

Parameters

audio	the AgsAudio	
n_params	the count of elements in following arrays	
parameter_name	the parameter's name array	
value	the value array	

Since: 3.0.0

ags_audio_recursive_run_stage ()

```
void
ags_audio_recursive_run_stage (AgsAudio *audio,
                               gint sound_scope,
                               guint staging_flags);
```

Resets *audio* 's *sound_scope* specified by *staging_flags* .

Parameters

audio	the AgsAudio object	
sound_scope	the scope to reset	
staging_flags	the stage to enable	

Since: 3.0.0

ags_audio_new ()

```
AgsAudio~*
ags_audio_new (GObject *output_soundcard);
```

Creates an AgsAudio, with defaults of *output_soundcard* .

Parameters

output_soundcard	the AgsSoundcard to use for output	

Returns

a new AgsAudio

Since: 3.0.0

AGS_AUDIO()

```
#define AGS_AUDIO(obj)                  (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_AUDIO,  ↩
    AgsAudio))
```

AGS_AUDIO_CLASS()

```
#define AGS_AUDIO_CLASS(class)          (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_AUDIO,  ↩
    AgsAudioClass))
```

AGS_AUDIO_GET_CLASS()

```
#define AGS_AUDIO_GET_CLASS(obj)        (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_AUDIO,  ↩
    AgsAudioClass))
```

AGS_IS_AUDIO()

```
#define AGS_IS_AUDIO(obj)               (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_AUDIO))
```

AGS_IS_AUDIO_CLASS()

```
#define AGS_IS_AUDIO_CLASS(class)       (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_AUDIO))
```

ags_audio_flags_get_type ()

```
GType
ags_audio_flags_get_type ();
```

ags_audio_get_type ()

```
GType
ags_audio_get_type ();
```

Types and Values

enum AgsAudioFlags

Enum values to control the behavior or indicate internal state of AgsAudio by enable/disable as flags.

Members

AGS_AUDIO_NO_OUTPUT	no output provided
AGS_AUDIO_NO_INPUT	no input provided
AGS_AUDIO_SYNC	input/output is mapped synchronously
AGS_AUDIO_ASYNC	input/output is mapped asynchronously
AGS_AUDIO_OUTPUT_HAS_RECYCLING	output has recycling
AGS_AUDIO_OUTPUT_HAS_SYNTH	output has synth
AGS_AUDIO_INPUT_HAS_RECYCLING	input has recycling
AGS_AUDIO_INPUT_HAS_SYNTH	input has synth
AGS_AUDIO_INPUT_HAS_FILE	input takes file
AGS_AUDIO_CAN_NEXT_ACTIVE	dispatch can next active
AGS_AUDIO_SKIP_OUTPUT	skip output as processing audio data

AGS_AUDIO_SKIP_INPUT	skip input as processing audio data
AGS_AUDIO_BYPASS	don't apply any data

AGS_TYPE_AUDIO

```
#define AGS_TYPE_AUDIO                (ags_audio_get_type ())
```

struct AgsAudio

```
struct AgsAudio;
```

struct AgsAudioClass

```
struct AgsAudioClass {
  GObjectClass gobject;

  void (*set_audio_channels)(AgsAudio *audio,
           guint audio_channels, guint audio_channels_old);
  void (*set_pads)(AgsAudio *audio,
       GType channel_type,
       guint pads, guint pads_old);

  void (*duplicate_recall)(AgsAudio *audio,
         AgsRecallID *recall_id,
         guint pad, guint audio_channel,
         guint line);
  void (*resolve_recall)(AgsAudio *audio,
       AgsRecallID *recall_id);

  void (*init_recall)(AgsAudio *audio,
         AgsRecallID *recall_id, guint staging_flags);
  void (*play_recall)(AgsAudio *audio,
         AgsRecallID *recall_id, guint staging_flags);

  void (*done_recall)(AgsAudio *audio,
         AgsRecallID *recall_id);
  void (*cancel_recall)(AgsAudio *audio,
      AgsRecallID *recall_id);

  void (*cleanup_recall)(AgsAudio *audio,
       AgsRecallID *recall_id);

  GList* (*start)(AgsAudio *audio,
```

```
      gint sound_scope);
  void (*stop)(AgsAudio *audio,
         GList *recall_id, gint sound_scope);

  GList* (*check_scope)(AgsAudio *audio, gint sound_scope);

  void (*recursive_run_stage)(AgsAudio *audio,
            gint sound_scope, guint staging_flags);
};
```

Property Details

The "absolute-key" property

```
  "absolute-key"              int
```

The absolute key lower.

Owner: AgsAudio

Flags: Read / Write

Allowed values: [0,128]

Default value: 0

Since: 3.0.0

The "audio-channels" property

```
  "audio-channels"            guint
```

The audio channels count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "audio-end-mapping" property

```
  "audio-end-mapping"         guint
```

The audio end mapping.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "audio-name" property

```
"audio-name"                char~*
```

The name of audio object.

Owner: AgsAudio

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "audio-start-mapping" property

```
"audio-start-mapping"       guint
```

The audio start mapping.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "automation" property

```
"automation"                gpointer
```

The AgsAutomation it contains.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "bpm" property

```
"bpm"                       double
```

The bpm.

Owner: AgsAudio

Flags: Read / Write

Allowed values: >= 0

Default value: 120

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer length.

Owner: AgsAudio

Flags: Read / Write

Default value: 1024

Since: 3.0.0

The "cursor" property

```
"cursor"                     gpointer
```

The GObject implementing AgsCursor interface.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "denominator" property

```
"denominator"                guint
```

The denominator of time signature.

Owner: AgsAudio

Flags: Read / Write

Allowed values: <= 32

Default value: 4

Since: 3.0.0

The "format" property

```
"format"                     guint
```

The format.

Owner: AgsAudio

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "input" property

```
"input"                    AgsInput~*
```

The AgsInput it contains.

Owner: AgsAudio

Flags: Read

Since: 3.0.0

The "input-audio-file" property

```
"input-audio-file"         GObject~*
```

The assigned AgsAudioFile acting as default sink.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "input-lines" property

```
"input-lines"              guint
```

The input lines count.

Owner: AgsAudio

Flags: Read

Default value: 0

Since: 3.0.0

The "input-midi-file" property

```
"input-midi-file"          GObject~*
```

The assigned AgsMidiFile acting as default sink.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "input-pads" property

```
"input-pads"               guint
```

The input pads count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "input-sequencer" property

```
"input-sequencer"          GObject~*
```

The assigned AgsSequencer acting as default source.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "input-soundcard" property

```
"input-soundcard"          GObject~*
```

The assigned AgsSoundcard acting as default source.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "is-minor" property

```
"is-minor"                 gboolean
```

Is minor.

Owner: AgsAudio

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "key" property

```
"key"                      guint
```

The key relative to octave.

Owner: AgsAudio

Flags: Read / Write

Allowed values: <= 12

Default value: 0

Since: 3.0.0

The "loop-end" property

```
"loop-end"                 guint64
```

The audio's loop end.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "loop-start" property

```
"loop-start"                guint64
```

The audio's loop start.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "max-audio-channels" property

```
"max-audio-channels"        guint
```

The maximum audio channels count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "max-input-pads" property

```
"max-input-pads"            guint
```

The maximum input pads count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "max-output-pads" property

```
"max-output-pads"           guint
```

The maximum output pads count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "midi" property

```
"midi"                      gpointer
```

The AgsMidi it contains.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "midi-channel" property

```
"midi-channel"              guint
```

The midi channel.

Owner: AgsAudio

Flags: Read / Write

Allowed values: <= 16

Default value: 0

Since: 3.0.0

The "midi-end-mapping" property

```
"midi-end-mapping"          guint
```

The midi end mapping.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "midi-start-mapping" property

```
"midi-start-mapping"        guint
```

The midi start mapping.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "min-audio-channels" property

```
"min-audio-channels"        guint
```

The minimum audio channels count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "min-input-pads" property

```
"min-input-pads"          guint
```

The minimum input pads count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "min-output-pads" property

```
"min-output-pads"         guint
```

The minimum output pads count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "notation" property

```
"notation"                gpointer
```

The AgsNotation it contains.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "numerator" property

```
"numerator"               guint
```

The numerator of time signature.

Owner: AgsAudio

Flags: Read / Write

Allowed values: <= 32

Default value: 4

Since: 3.0.0

The "octave" property

```
"octave"                    int
```

The octave lower.

Owner: AgsAudio

Flags: Read / Write

Allowed values: [0,10]

Default value: 0

Since: 3.0.0

The "offset" property

```
"offset"                    guint64
```

The audio's offset.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "output" property

```
"output"                    AgsOutput~*
```

The AgsOutput it contains.

Owner: AgsAudio

Flags: Read

Since: 3.0.0

The "output-audio-file" property

```
"output-audio-file"         GObject~*
```

The assigned AgsAudioFile acting as default sink.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "output-lines" property

```
"output-lines"              guint
```

The output lines count.

Owner: AgsAudio

Flags: Read

Default value: 0

Since: 3.0.0

The "output-midi-file" property

```
"output-midi-file"          GObject~*
```

The assigned AgsMidiFile acting as default sink.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "output-pads" property

```
"output-pads"               guint
```

The output pads count.

Owner: AgsAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "output-sequencer" property

```
"output-sequencer"          GObject~*
```

The assigned AgsSequencer acting as default source.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "output-soundcard" property

```
"output-soundcard"          GObject~*
```

The assigned AgsSoundcard acting as default sink.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "play" property

```
"play"                      gpointer
```

The AgsRecall it contains in play-context.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "playback-domain" property

```
"playback-domain"          AgsPlaybackDomain~*
```

The assigned AgsPlaybackDomain.

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "preset" property

```
"preset"                   gpointer
```

The assigned GList containing AgsPreset information.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "recall" property

```
"recall"                   gpointer
```

The AgsRecall it contains in recall-context.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "recall-container" property

```
"recall-container"         gpointer
```

The AgsRecallContainer it contains in container-context.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "recall-id" property

```
"recall-id"                gpointer
```

The assigned AgsRecallID.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "recycling-context" property

```
"recycling-context"        gpointer
```

The assigned AgsRecyclingContext.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
"samplerate"               guint
```

The samplerate.

Owner: AgsAudio

Flags: Read / Write

Default value: 48000

Since: 3.0.0

The "sf2-synth-generator" property

```
"sf2-synth-generator"      gpointer
```

The assigned GList containing AgsSF2SynthGenerator information.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.4.0

The "sfz-synth-generator" property

```
"sfz-synth-generator"      gpointer
```

The assigned GList containing AgsSFZSynthGenerator information.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.4.0

The "sharp-flats" property

```
"sharp-flats"                guint
```

The sharp/flats count.

Owner: AgsAudio

Flags: Read / Write

Allowed values: <= 12

Default value: 0

Since: 3.0.0

The "synth-generator" property

```
"synth-generator"            gpointer
```

The assigned GList containing AgsSynthGenerator information.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

The "time-signature" property

```
"time-signature"             char~*
```

The time signature.

Owner: AgsAudio

Flags: Read

Default value: "4/4"

Since: 3.0.0

The "wave" property

```
"wave"                       gpointer
```

The AgsWave it contains.

[transfer full]

Owner: AgsAudio

Flags: Read / Write

Since: 3.0.0

Signal Details

The "cancel-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               gpointer  user_data)
```

The ::cancel-recall signal notifies about canceld recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "check-scope" signal

```
gpointer
user_function (AgsAudio *audio,
               int       sound_scope,
               gpointer  user_data)
```

The ::check-scope method returns the appropriate recall id of *sound_scope* .

Parameters

audio	the AgsAudio to check the scopes	
sound_scope	the affected scope	
user_data	user data set when the signal handler was connected.	

Returns

the GList containing AgsRecallID

Flags: Run Last

Since: 3.0.0

The "cleanup-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               gpointer  user_data)
```

The ::cleanup-recall signal notifies about cleanup recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "done-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               gpointer  user_data)
```

The ::done-recall signal notifies about doned recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "duplicate-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               guint     pad,
               guint     audio_channel,
               guint     line,
               gpointer  user_data)
```

The ::duplicate-recall signal notifies about duplicated recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
pad	the pad	
audio_channel	the audio channel	
line	the line	

user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "init-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               guint     staging_flags,
               gpointer  user_data)
```

The ::init-recall signal notifies about initd recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
staging_flags	the staging flags	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "play-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               guint     staging_flags,
               gpointer  user_data)
```

The ::play-recall signal notifies about playd recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
staging_flags	the staging flags	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "recursive-run-stage" signal

```
void
user_function (AgsAudio *audio,
               int       sound_scope,
               guint     staging_flags,
               gpointer  user_data)
```

The ::recursive-run-stage signal is invoked while run staging of *audio* for *sound_scope* .

Parameters

audio	the AgsAudio to run	
sound_scope	the affected scope	
staging_flags	the flags to set	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "resolve-recall" signal

```
void
user_function (AgsAudio *audio,
               GObject  *recall_id,
               gpointer  user_data)
```

The ::resolve-recall signal notifies about resolved recalls.

Parameters

audio	the AgsAudio	
recall_id	the AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "set-audio-channels" signal

```
void
user_function (AgsAudio *audio,
               guint     audio_channels_new,
               guint     audio_channels_old,
               gpointer  user_data)
```

The ::set-audio-channels signal notifies about changes in channel alignment.

Parameters

audio	the object to adjust the channels.	
audio_channels_new	new audio channel count	
audio_channels_old	old audio channel count	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "set-pads" signal

```
void
user_function (AgsAudio *audio,
               gpointer  channel_type,
               guint     pads_new,
               guint     pads_old,
               gpointer  user_data)
```

The ::set-pads signal notifies about changes in channel alignment.

Parameters

audio	the object to adjust pads.	
channel_type	either AGS_TYPE_INPUT or AGS_TYPE_OUTPUT	
pads_new	new pad count	
pads_old	old pad count	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "start" signal

```
gpointer
user_function (AgsAudio *audio,
               int       sound_scope,
               gpointer  user_data)
```

The ::start signal is invoked while starting playback of *audio* .

Parameters

audio	the AgsAudio starts playing	
sound_scope	the affected scope	
user_data	user data set when the signal handler was connected.	

Returns

the GList containing AgsRecallID

Flags: Run Last

Since: 3.0.0

The "stop" signal

```
void
user_function (AgsAudio *audio,
               gpointer  recall_id,
               int       sound_scope,
               gpointer  user_data)
```

The ::stop signal is invoked while stoping playback of *audio* .

Parameters

audio	the AgsAudio stops playing	
recall_id	a GList containing AgsRecallID	
sound_scope	the affected scope	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

9.4 AgsChannel

AgsChannel — A channel is a node within the audio processing tree and does recycling

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CHANNEL_GET_OBJ_MUTEX()
#define	AGS_CHANNEL_GET_PLAY_MUTEX()
#define	AGS_CHANNEL_GET_RECALL_MUTEX()
GQuark	ags_channel_error_quark ()
GRecMutex *	ags_channel_get_obj_mutex ()
GRecMutex *	ags_channel_get_play_mutex ()
GRecMutex *	ags_channel_get_recall_mutex ()
gboolean	ags_channel_test_flags ()
void	ags_channel_set_flags ()
void	ags_channel_unset_flags ()
gboolean	ags_channel_test_ability_flags ()
void	ags_channel_set_ability_flags ()
void	ags_channel_unset_ability_flags ()
gboolean	ags_channel_test_behaviour_flags ()
void	ags_channel_set_behaviour_flags ()
void	ags_channel_unset_behaviour_flags ()
gboolean	ags_channel_test_staging_flags ()
void	ags_channel_set_staging_flags ()
void	ags_channel_unset_staging_flags ()
gboolean	ags_channel_test_staging_completed ()
void	ags_channel_set_staging_completed ()
void	ags_channel_unset_staging_completed ()
GObject *	ags_channel_get_audio ()
void	ags_channel_set_audio ()
AgsChannel *	ags_channel_next ()
AgsChannel *	ags_channel_prev ()
AgsChannel *	ags_channel_next_pad ()
AgsChannel *	ags_channel_prev_pad ()
AgsChannel *	ags_channel_first ()
AgsChannel *	ags_channel_last ()
AgsChannel *	ags_channel_nth ()
AgsChannel *	ags_channel_pad_first ()
AgsChannel *	ags_channel_pad_last ()
AgsChannel *	ags_channel_pad_nth ()
AgsChannel *	ags_channel_first_with_recycling ()
AgsChannel *	ags_channel_last_with_recycling ()
AgsChannel *	ags_channel_prev_with_recycling ()
AgsChannel *	ags_channel_next_with_recycling ()
AgsChannel *	ags_channel_get_link ()
void	ags_channel_set_link ()
void	ags_channel_reset_recycling ()
void	ags_channel_recycling_changed ()
GObject *	ags_channel_get_output_soundcard ()
void	ags_channel_set_output_soundcard ()
gint	ags_channel_get_output_soundcard_channel ()
void	ags_channel_set_output_soundcard_channel ()
GObject *	ags_channel_get_input_soundcard ()
void	ags_channel_set_input_soundcard ()
gint	ags_channel_get_input_soundcard_channel ()
void	ags_channel_set_input_soundcard_channel ()
guint	ags_channel_get_samplerate ()
void	ags_channel_set_samplerate ()
guint	ags_channel_get_buffer_size ()
void	ags_channel_set_buffer_size ()

guint	ags_channel_get_format ()
void	ags_channel_set_format ()
guint	ags_channel_get_pad ()
void	ags_channel_set_pad ()
guint	ags_channel_get_audio_channel ()
void	ags_channel_set_audio_channel ()
guint	ags_channel_get_line ()
void	ags_channel_set_line ()
gint	ags_channel_get_octave ()
void	ags_channel_set_octave ()
guint	ags_channel_get_key ()
void	ags_channel_set_key ()
gint	ags_channel_get_absolute_key ()
void	ags_channel_set_absolute_key ()
GList *	ags_channel_get_pattern ()
void	ags_channel_set_pattern ()
void	ags_channel_add_pattern ()
void	ags_channel_remove_pattern ()
GObject *	ags_channel_get_playback ()
void	ags_channel_set_playback ()
GList *	ags_channel_get_recall_id ()
void	ags_channel_set_recall_id ()
void	ags_channel_add_recall_id ()
void	ags_channel_remove_recall_id ()
GList *	ags_channel_get_recall_container ()
void	ags_channel_set_recall_container ()
void	ags_channel_add_recall_container ()
void	ags_channel_remove_recall_container ()
GList *	ags_channel_get_play ()
void	ags_channel_set_play ()
GList *	ags_channel_get_recall ()
void	ags_channel_set_recall ()
void	ags_channel_add_recall ()
void	ags_channel_insert_recall ()
void	ags_channel_remove_recall ()
void	ags_channel_duplicate_recall ()
void	ags_channel_resolve_recall ()
void	ags_channel_init_recall ()
void	ags_channel_play_recall ()
void	ags_channel_done_recall ()
void	ags_channel_cancel_recall ()
void	ags_channel_cleanup_recall ()
GList *	ags_channel_start ()
void	ags_channel_stop ()
GList *	ags_channel_check_scope ()
GList *	ags_channel_collect_all_channel_ports ()
GList *	ags_channel_collect_all_channel_ports_by_specifier_and_context ()
AgsChannel *	ags_channel_get_level ()
void	ags_channel_recursive_set_property ()
void	ags_channel_recursive_run_stage ()
AgsChannel *	ags_channel_new ()
#define	AGS_CHANNEL()
#define	AGS_CHANNEL_CLASS()
#define	AGS_CHANNEL_GET_CLASS()
#define	AGS_IS_CHANNEL()
#define	AGS_IS_CHANNEL_CLASS()
GType	ags_channel_flags_get_type ()

GType	ags_channel_get_type ()

Properties

int	absolute-key	Read / Write
AgsAudio *	audio	Read / Write
guint	audio-channel	Read
guint	buffer-size	Read / Write
AgsRecycling *	first-recycling	Read
guint	format	Read / Write
GObject *	input-soundcard	Read / Write
int	input-soundcard-channel	Read / Write
guint	key	Read / Write
AgsRecycling *	last-recycling	Read
guint	line	Read
AgsChannel *	link	Read / Write
guint	midi-note	Read / Write
AgsChannel *	next	Read
AgsChannel *	next-pad	Read
double	note-frequency	Read / Write
char *	note-key	Read / Write
int	octave	Read / Write
GObject *	output-soundcard	Read / Write
int	output-soundcard-channel	Read / Write
guint	pad	Read
gpointer	pattern	Read / Write
gpointer	play	Read / Write
AgsPlayback *	playback	Read / Write
AgsChannel *	prev	Read
AgsChannel *	prev-pad	Read
gpointer	recall	Read / Write
gpointer	recall-container	Read / Write
gpointer	recall-id	Read / Write
gpointer	recycling-context	Read / Write
guint	samplerate	Read / Write

Signals

void	cancel-recall	Run Last
gpointer	check-scope	Run Last
void	cleanup-recall	Run Last
void	done-recall	Run Last
void	duplicate-recall	Run Last
void	init-recall	Run Last
void	play-recall	Run Last
void	recursive-run-stage	Run Last
void	recycling-changed	Run Last
void	resolve-recall	Run Last
gpointer	start	Run Last
void	stop	Run Last

Types and Values

#define	AGS_CHANNEL_MINIMUM_OCTAVE
#define	AGS_CHANNEL_MAXIMUM_OCTAVE
#define	AGS_CHANNEL_DEFAULT_OCTAVE
#define	AGS_CHANNEL_MINIMUM_OCTAVE_SEMITONE
#define	AGS_CHANNEL_MAXIMUM_OCTAVE_SEMITONE
#define	AGS_CHANNEL_DEFAULT_OCTAVE_SEMITONE
#define	AGS_CHANNEL_MINIMUM_SEMITONE
#define	AGS_CHANNEL_MAXIMUM_SEMITONE
#define	AGS_CHANNEL_DEFAULT_SEMITONE
#define	AGS_CHANNEL_MINIMUM_NOTE_FREQUENCY
#define	AGS_CHANNEL_MAXIMUM_NOTE_FREQUENCY
#define	AGS_CHANNEL_DEFAULT_NOTE_FREQUENCY
#define	AGS_CHANNEL_MINIMUM_MIDI_NOTE
#define	AGS_CHANNEL_MAXIMUM_MIDI_NOTE
#define	AGS_CHANNEL_DEFAULT_MIDI_NOTE
enum	AgsChannelFlags
#define	AGS_CHANNEL_ERROR
enum	AgsChannelError
#define	AGS_TYPE_CHANNEL
struct	AgsChannel
struct	AgsChannelClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsChannelFlags
GObject
&#x2570;&#x2500;&#x2500; AgsChannel
    &#x251c;&#x2500;&#x2500; AgsInput
    &#x2570;&#x2500;&#x2500; AgsOutput
```

Implemented Interfaces

AgsChannel implements AgsConnectable.

Includes

```
#include <ags/audio/ags_channel.h>
```

Description

AgsChannel is the entry point to the entire audio tree and its nested recycling tree.

Every channel has its own AgsRecallID. As modifying link a new AgsRecyclingContext is indicated, since it acts as a kind of recall id tree context.

It hosts AgsPattern to implement step sequencers.

Make sure you have set the matching AgsSoundAbilityFlags flags using ags_channel_set_ability_flags(). This is going to setup threads per matching AgsSoundScope scope.

Multi-threaded processing per output line only applies if AGS_PLAYBACK_SUPER_THREADED_CHANNEL is set on matching AgsPlayback.

You might want to listen to changes in the nested recycling tree, too. ags_channel_recycling_changed() notifies about changes of processing structure in the deep audio tree.

The `first-recycling` and `last-recycling` property gives you the boundaries of processing structure in the audio tree.

If channel is an AgsOutput and is a signal source i.e. instrument or such `first-recycling` and `last-recycling` point to the very same AgsRecycling. In this case it is never altered.

On the other hand ags_channel_set_link() may alter your nested recycling tree. It may change the recycling context.

There are common functions to iterate channels.

There is a playback and recall context for hosting AgsRecall implementations. Usually you instantiate a recall for both contices. See `play` and `recall` properties.

Functions

AGS_CHANNEL_GET_OBJ_MUTEX()

```
#define AGS_CHANNEL_GET_OBJ_MUTEX(obj) (&(((AgsChannel *) obj)->obj_mutex))
```

AGS_CHANNEL_GET_PLAY_MUTEX()

```
#define AGS_CHANNEL_GET_PLAY_MUTEX(obj) (&(((AgsChannel *) obj)->play_mutex))
```

AGS_CHANNEL_GET_RECALL_MUTEX()

```
#define AGS_CHANNEL_GET_RECALL_MUTEX(obj) (&(((AgsChannel *) obj)->recall_mutex))
```

ags_channel_error_quark ()

```
GQuark
ags_channel_error_quark ();
```

ags_channel_get_obj_mutex ()

```
GRecMutex~*
ags_channel_get_obj_mutex (AgsChannel *channel);
```

Get object mutex.

Parameters

channel	the AgsChannel

Returns

the GRecMutex to lock *channel*

Since: 3.1.0

ags_channel_get_play_mutex ()

```
GRecMutex~*
ags_channel_get_play_mutex (AgsChannel *channel);
```

Get play mutex.

Parameters

channel	the AgsChannel

Returns

the GRecMutex to lock *channel* 's play property

Since: 3.1.0

ags_channel_get_recall_mutex ()

```
GRecMutex~*
ags_channel_get_recall_mutex (AgsChannel *channel);
```

Get recall mutex.

Parameters

channel	the AgsChannel

Returns

the GRecMutex to lock *channel* 's recall property

Since: 3.1.0

ags_channel_test_flags ()

```
gboolean
ags_channel_test_flags (AgsChannel *channel,
                        guint flags);
```

Test *flags* to be set on *channel* .

Parameters

channel	the AgsChannel
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_channel_set_flags ()

```
void
ags_channel_set_flags (AgsChannel *channel,
                       guint flags);
```

Enable a feature of *channel* .

Parameters

channel	the AgsChannel	
flags	see AgsChannelFlags	

Since: 3.0.0

ags_channel_unset_flags ()

```
void
ags_channel_unset_flags (AgsChannel *channel,
                         guint flags);
```

Disable a feature of *channel* .

Parameters

channel	the AgsChannel	
flags	see AgsChannelFlags	

Since: 3.0.0

ags_channel_test_ability_flags ()

```
gboolean
ags_channel_test_ability_flags (AgsChannel *channel,
                                guint ability_flags);
```

Test *ability_flags* to be set on *channel* .

Parameters

channel	the AgsChannel	
ability_flags	the ability flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_channel_set_ability_flags ()

```
void
ags_channel_set_ability_flags (AgsChannel *channel,
                               guint ability_flags);
```

Enable an ability of AgsChannel.

Parameters

channel	the AgsChannel	
ability_flags	see enum AgsSoundAbilityFlags	

Since: 3.0.0

ags_channel_unset_ability_flags ()

```
void
ags_channel_unset_ability_flags (AgsChannel *channel,
                                 guint ability_flags);
```

Disable an ability of AgsChannel.

Parameters

channel	the AgsChannel	
ability_flags	see enum AgsSoundAbilityFlags	

Since: 3.0.0

ags_channel_test_behaviour_flags ()

```
gboolean
ags_channel_test_behaviour_flags (AgsChannel *channel,
                                  guint behaviour_flags);
```

Test *behaviour_flags* to be set on *channel* .

Parameters

channel	the AgsChannel	
behaviour_flags	the behaviour flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_channel_set_behaviour_flags ()

```
void
ags_channel_set_behaviour_flags (AgsChannel *channel,
                                 guint behaviour_flags);
```

Set behaviour flags.

Parameters

channel	the AgsChannel	
behaviour_flags	the behaviour flags	

Since: 3.0.0

ags_channel_unset_behaviour_flags ()

```
void
ags_channel_unset_behaviour_flags (AgsChannel *channel,
                                   guint behaviour_flags);
```

Unset behaviour flags.

Parameters

channel	the AgsChannel	
behaviour_flags	the behaviour flags	

Since: 3.0.0

ags_channel_test_staging_flags ()

```
gboolean
ags_channel_test_staging_flags (AgsChannel *channel,
                                gint sound_scope,
                                guint staging_flags);
```

Test *staging_flags* to be set on *channel* .

Parameters

channel	the AgsChannel	
sound_scope	the AgsSoundScope to test	
staging_flags	the staging flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_channel_set_staging_flags ()

```
void
ags_channel_set_staging_flags (AgsChannel *channel,
                               gint sound_scope,
                               guint staging_flags);
```

Set staging flags.

Parameters

channel	the AgsChannel	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	
staging_flags	the staging flags	

Since: 3.0.0

ags_channel_unset_staging_flags ()

```
void
ags_channel_unset_staging_flags (AgsChannel *channel,
                                 gint sound_scope,
                                 guint staging_flags);
```

Unset staging flags.

Parameters

channel	the AgsChannel	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	
staging_flags	the staging flags	

Since: 3.0.0

ags_channel_test_staging_completed ()

```
gboolean
ags_channel_test_staging_completed (AgsChannel *channel,
                                    gint sound_scope);
```

Test *sound_scope* to be completed on *channel* .

Parameters

channel	the AgsChannel	
sound_scope	the AgsSoundScope to test	

Returns

TRUE if completed, otherwise FALSE

Since: 3.3.0

ags_channel_set_staging_completed ()

```
void
ags_channel_set_staging_completed (AgsChannel *channel,
                                   gint sound_scope);
```

Set *sound_scope* to be completed.

Parameters

channel	the AgsChannel	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	

Since: 3.3.0

ags_channel_unset_staging_completed ()

```
void
ags_channel_unset_staging_completed (AgsChannel *channel,
                                     gint sound_scope);
```

Unset *sound_scope* to be completed.

Parameters

channel	the AgsChannel	
sound_scope	the AgsSoundScope to apply, or -1 to apply to all	

Since: 3.3.0

ags_channel_get_audio ()

```
GObject~*
ags_channel_get_audio (AgsChannel *channel);
```

Get audio.

Parameters

channel	the AgsChannel	

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_channel_set_audio ()

```
void
ags_channel_set_audio (AgsChannel *channel,
                       GObject *audio);
```

Set audio.

Parameters

channel	the AgsChannel	
audio	the AgsAudio	

Since: 3.1.0

ags_channel_next ()

```
AgsChannel~*
ags_channel_next (AgsChannel *channel);
```

Iterate *channel* .

Parameters

channel	the AgsChannel	

Returns

the next of AgsChannel if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_channel_prev ()

```
AgsChannel~*
ags_channel_prev (AgsChannel *channel);
```

Iterate *channel* .

Parameters

channel	the AgsChannel	

Returns

the prev of AgsChannel if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_channel_next_pad ()

```
AgsChannel~*
ags_channel_next_pad (AgsChannel *channel);
```

Iterate *channel* .

Parameters

channel	the AgsChannel

Returns

the next pad of AgsChannel if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_channel_prev_pad ()

```
AgsChannel~*
ags_channel_prev_pad (AgsChannel *channel);
```

Iterate *channel* .

Parameters

channel	the AgsChannel

Returns

the prev pad of AgsChannel if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_channel_first ()

```
AgsChannel~*
ags_channel_first (AgsChannel *channel);
```

Iterates until the first AgsChannel was found.

Parameters

channel	an AgsChannel

Returns

the first AgsChannel.

[transfer full]

Since: 3.0.0

ags_channel_last ()

```
AgsChannel~*
ags_channel_last (AgsChannel *channel);
```

Iterates until the last AgsChannel was found.

Parameters

channel	an AgsChannel

Returns

the last AgsChannel.

[transfer full]

Since: 3.0.0

ags_channel_nth ()

```
AgsChannel~*
ags_channel_nth (AgsChannel *channel,
                 guint nth);
```

Iterates `nth` times forward.

Parameters

channel	an AgsChannel
nth	the count to iterate

Returns

the nth AgsChannel.

[transfer full]

Since: 3.0.0

ags_channel_pad_first ()

```
AgsChannel~*
ags_channel_pad_first (AgsChannel *channel);
```

Iterates until the first pad has been reached.

Parameters

channel	an AgsChannel

Returns

the first AgsChannel with the same audio channel as `channel` .

[transfer full]

Since: 3.0.0

ags_channel_pad_last ()

```
AgsChannel~*
ags_channel_pad_last (AgsChannel *channel);
```

Iterates until the last pad has been reached.

Parameters

channel	an AgsChannel

Returns

the last AgsChannel with the same audio channel as `channel` .

[transfer full]

Since: 3.0.0

ags_channel_pad_nth ()

```
AgsChannel~*
ags_channel_pad_nth (AgsChannel *channel,
                     guint nth);
```

Iterates nth times.

Parameters

channel	an AgsChannel
nth	the count of pads to step

Returns

the nth pad.

[transfer full]

Since: 3.0.0

ags_channel_first_with_recycling ()

```
AgsChannel~*
ags_channel_first_with_recycling (AgsChannel *channel);
```

Find first recycling related to `channel` .

Parameters

channel	an AgsChannel

Returns

the first channel with an AgsRecycling.

[transfer full]

Since: 3.0.0

ags_channel_last_with_recycling ()

```
AgsChannel~*
ags_channel_last_with_recycling (AgsChannel *channel);
```

Find last recycling related to *channel* .

Parameters

channel	an AgsChannel

Returns

the last channel with an AgsRecycling.

[transfer full]

Since: 3.0.0

ags_channel_prev_with_recycling ()

```
AgsChannel~*
ags_channel_prev_with_recycling (AgsChannel *channel);
```

Find previous AgsRecycling of *channel* .

Parameters

channel	an AgsChannel

Returns

the prev channel with an AgsRecycling.

[transfer full]

Since: 3.0.0

ags_channel_next_with_recycling ()

```
AgsChannel~*
ags_channel_next_with_recycling (AgsChannel *channel);
```

Find next AgsRecycling of *channel* .

Parameters

channel	an AgsChannel

Returns

the next channel with an AgsRecycling.

[transfer full]

Since: 3.0.0

ags_channel_get_link ()

```
AgsChannel~*
ags_channel_get_link (AgsChannel *channel);
```

Ascend/descend `channel` .

Parameters

channel	the AgsChannel

Returns

the link of AgsChannel if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_channel_set_link ()

```
void
ags_channel_set_link (AgsChannel *channel,
                      AgsChannel *link,
                      GError **error);
```

Change the linking of AgsChannel objects. Sets link, calls ags_channel_reset_recycling(). Further it does loop detection and makes your machine running.

Parameters

channel	an AgsChannel to link	
link	an other AgsChannel to link with	
error	you may retrieve a AGS_CHANNEL_ERROR_LOOP_IN_LINK error	

Since: 3.0.0

ags_channel_reset_recycling ()

```
void
ags_channel_reset_recycling (AgsChannel *channel,
                             AgsRecycling *first_recycling,
                             AgsRecycling *last_recycling);
```

Called by ags_channel_set_link() to handle outdated AgsRecycling references. Invoke only by a task.

Parameters

channel	the channel to reset	
first_recycling	the recycling to set for channel->first_recycling	
last_recycling	the recycling to set for channel->last_recycling	

Since: 3.0.0

ags_channel_recycling_changed ()

```
void
ags_channel_recycling_changed (AgsChannel *channel,
                               AgsRecycling *old_start_region,
                               AgsRecycling *old_end_region,
                               AgsRecycling *new_start_region,
                               AgsRecycling *new_end_region,
                               AgsRecycling *old_start_changed_region,
                               AgsRecycling *old_end_changed_region,
                               AgsRecycling *new_start_changed_region,
                               AgsRecycling *new_end_changed_region);
```

Modify recycling. Asynchronously only.

Parameters

channel	the object recycling changed	
old_start_region	first recycling	
old_end_region	last recycling	
new_start_region	new first recycling	
new_end_region	new last recycling	
old_start_changed_region	modified link recycling start	
old_end_changed_region	modified link recyclig end	
new_start_changed_region	replacing link recycling start	
new_end_changed_region	replacing link recycling end	

Since: 3.0.0

ags_channel_get_output_soundcard ()

```
GObject~*
ags_channel_get_output_soundcard (AgsChannel *channel);
```

Get the output soundcard object of *channel* .

Parameters

channel	the AgsChannel	

Returns

the output soundcard.

[transfer full]

Since: 3.1.0

ags_channel_set_output_soundcard ()

```
void
ags_channel_set_output_soundcard (AgsChannel *channel,
                                  GObject *output_soundcard);
```

Set the output soundcard object of `channel` .

Parameters

channel	an AgsChannel	
output_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_channel_get_output_soundcard_channel ()

```
gint
ags_channel_get_output_soundcard_channel
                                (AgsChannel *channel);
```

Get the output soundcard object of `channel` .

Parameters

channel	the AgsChannel	

Returns

the output soundcard

Since: 4.0.0

ags_channel_set_output_soundcard_channel ()

```
void
ags_channel_set_output_soundcard_channel
                                (AgsChannel *channel,
                                 gint output_soundcard_channel);
```

Set the output soundcard channel of `channel` .

Parameters

channel	an AgsChannel	
output_soundcard_channel	the output soundcard channel	

Since: 4.0.0

ags_channel_get_input_soundcard ()

```
GObject~*
ags_channel_get_input_soundcard (AgsChannel *channel);
```

Get the input soundcard object of *channel* .

Parameters

channel	the AgsChannel	

Returns

the input soundcard.

[transfer full]

Since: 3.1.0

ags_channel_set_input_soundcard ()

```
void
ags_channel_set_input_soundcard (AgsChannel *channel,
                                 GObject *input_soundcard);
```

Set the input soundcard object of *channel* .

Parameters

channel	an AgsChannel	
input_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_channel_get_input_soundcard_channel ()

```
gint
ags_channel_get_input_soundcard_channel
                                (AgsChannel *channel);
```

Get the input soundcard object of *channel* .

Parameters

channel	the AgsChannel

Returns

the input soundcard

Since: 4.0.0

ags_channel_set_input_soundcard_channel ()

```
void
ags_channel_set_input_soundcard_channel
                                (AgsChannel *channel,
                                 gint input_soundcard_channel);
```

Set the input soundcard channel of *channel* .

Parameters

channel	an AgsChannel
input_soundcard_channel	the input soundcard channel

Since: 4.0.0

ags_channel_get_samplerate ()

```
guint
ags_channel_get_samplerate (AgsChannel *channel);
```

Gets samplerate.

Parameters

channel	the AgsChannel

Returns

the samplerate

Since: 3.1.0

ags_channel_set_samplerate ()

```
void
ags_channel_set_samplerate (AgsChannel *channel,
                            guint samplerate);
```

Set samplerate.

Parameters

channel	the AgsChannel	
samplerate	the samplerate	

Since: 3.0.0

ags_channel_get_buffer_size ()

```
guint
ags_channel_get_buffer_size (AgsChannel *channel);
```

Gets buffer size.

Parameters

channel	the AgsChannel	

Returns

the buffer size

Since: 3.1.0

ags_channel_set_buffer_size ()

```
void
ags_channel_set_buffer_size (AgsChannel *channel,
                             guint buffer_size);
```

Set buffer-size.

Parameters

channel	the AgsChannel	
buffer_size	the buffer_size	

Since: 3.0.0

ags_channel_get_format ()

```
guint
ags_channel_get_format (AgsChannel *channel);
```

Gets format.

Parameters

channel	the AgsChannel	

Returns

the format

Since: 3.1.0

ags_channel_set_format ()

```
void
ags_channel_set_format (AgsChannel *channel,
                        guint format);
```

Set format.

Parameters

channel	the AgsChannel	
format	the format	

Since: 3.0.0

ags_channel_get_pad ()

```
guint
ags_channel_get_pad (AgsChannel *channel);
```

Gets pad.

Parameters

channel	the AgsChannel	

Returns

the pad

Since: 3.1.0

ags_channel_set_pad ()

```
void
ags_channel_set_pad (AgsChannel *channel,
                     guint pad);
```

Set pad.

Parameters

channel	the AgsChannel	
pad	the pad	

Since: 3.0.0

ags_channel_get_audio_channel ()

```
guint
ags_channel_get_audio_channel (AgsChannel *channel);
```

Gets audio_channel.

Parameters

channel	the AgsChannel

Returns

the audio_channel

Since: 3.1.0

ags_channel_set_audio_channel ()

```
void
ags_channel_set_audio_channel (AgsChannel *channel,
                               guint audio_channel);
```

Set audio_channel.

Parameters

channel	the AgsChannel
audio_channel	the audio_channel

Since: 3.0.0

ags_channel_get_line ()

```
guint
ags_channel_get_line (AgsChannel *channel);
```

Gets line.

Parameters

channel	the AgsChannel

Returns

the line

Since: 3.1.0

ags_channel_set_line ()

```
void
ags_channel_set_line (AgsChannel *channel,
                      guint line);
```

Set line.

Parameters

channel	the AgsChannel	
line	the line	

Since: 3.0.0

ags_channel_get_octave ()

```
gint
ags_channel_get_octave (AgsChannel *channel);
```

Gets octave.

Parameters

channel	the AgsChannel	

Returns

the octave

Since: 3.1.0

ags_channel_set_octave ()

```
void
ags_channel_set_octave (AgsChannel *channel,
                        gint octave);
```

Sets octave.

Parameters

channel	the AgsChannel	
octave	the octave	

Since: 3.1.0

ags_channel_get_key ()

```
guint
ags_channel_get_key (AgsChannel *channel);
```

Gets key.

Parameters

channel	the AgsChannel

Returns

the key

Since: 3.1.0

ags_channel_set_key ()

```
void
ags_channel_set_key (AgsChannel *channel,
                     guint key);
```

Sets key.

Parameters

channel	the AgsChannel
key	the key

Since: 3.1.0

ags_channel_get_absolute_key ()

```
gint
ags_channel_get_absolute_key (AgsChannel *channel);
```

Gets absolute key.

Parameters

channel	the AgsChannel

Returns

the absolute key

Since: 3.1.0

ags_channel_set_absolute_key ()

```
void
ags_channel_set_absolute_key (AgsChannel *channel,
                              gint absolute_key);
```

Sets absolute key.

Parameters

channel	the AgsChannel	
absolute_key	the absolute key	

Since: 3.1.0

ags_channel_get_pattern ()

```
GList~*
ags_channel_get_pattern (AgsChannel *channel);
```

Get pattern.

Parameters

channel	the AgsChannel	

Returns

the GList containig AgsPattern.

[element-type AgsAudio.Pattern][transfer full]

Since: 3.1.0

ags_channel_set_pattern ()

```
void
ags_channel_set_pattern (AgsChannel *channel,
                         GList *pattern);
```

Set pattern by replacing existing.

Parameters

channel	the AgsChannel	
pattern	the GList containing AgsPattern.	*[element-type AgsAudio.Pattern][transfer full]*

Since: 3.1.0

ags_channel_add_pattern ()

```
void
ags_channel_add_pattern (AgsChannel *channel,
                         GObject *pattern);
```

Removes a pattern.

Parameters

channel	an AgsChannel	
pattern	the AgsPattern	

Since: 3.0.0

ags_channel_remove_pattern ()

```
void
ags_channel_remove_pattern (AgsChannel *channel,
                            GObject *pattern);
```

Removes a pattern.

Parameters

channel	an AgsChannel	
pattern	the AgsPattern	

Since: 3.0.0

ags_channel_get_playback ()

```
GObject~*
ags_channel_get_playback (AgsChannel *channel);
```

Get playback.

Parameters

channel	the AgsChannel	

Returns

the AgsPlayback.

[transfer full]

Since: 3.1.0

ags_channel_set_playback ()

```
void
ags_channel_set_playback (AgsChannel *channel,
                          GObject *playback);
```

Set playback.

Parameters

channel	the AgsChannel	
playback	the AgsPlayback	

Since: 3.1.0

ags_channel_get_recall_id ()

```
GList~*
ags_channel_get_recall_id (AgsChannel *channel);
```

Get recall id.

Parameters

channel	the AgsChannel	

Returns

the GList containig AgsRecallID.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.1.0

ags_channel_set_recall_id ()

```
void
ags_channel_set_recall_id (AgsChannel *channel,
                           GList *recall_id);
```

Set recall id by replacing existing.

Parameters

channel	the AgsChannel	
recall_id	the GList containing AgsRecallID.	*[element-type AgsAudio.RecallID][transfer full]*

Since: 3.1.0

ags_channel_add_recall_id ()

```
void
ags_channel_add_recall_id (AgsChannel *channel,
                           AgsRecallID *recall_id);
```

Adds a recall id.

Parameters

channel	an AgsChannel	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_channel_remove_recall_id ()

```
void
ags_channel_remove_recall_id (AgsChannel *channel,
                              AgsRecallID *recall_id);
```

Removes a recall id.

Parameters

channel	an AgsChannel	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_channel_get_recall_container ()

```
GList~*
ags_channel_get_recall_container (AgsChannel *channel);
```

Get recall_container.

Parameters

channel	the AgsChannel	

Returns

the GList containig AgsRecallContainer.

[element-type AgsAudio.RecallContainer][transfer full]

Since: 3.1.0

ags_channel_set_recall_container ()

```
void
ags_channel_set_recall_container (AgsChannel *channel,
                                  GList *recall_container);
```

Set recall_container by replacing existing.

Parameters

channel	the AgsChannel	
recall_container	the GList containing AgsRecallContainer.	*[element-type AgsAudio.RecallContainer][transfer full]*

Since: 3.1.0

ags_channel_add_recall_container ()

```
void
ags_channel_add_recall_container (AgsChannel *channel,
                                  GObject *recall_container);
```

Adds a recall container.

Parameters

channel	an AgsChannel	
recall_container	the AgsRecallContainer	

Since: 3.0.0

ags_channel_remove_recall_container ()

```
void
ags_channel_remove_recall_container (AgsChannel *channel,
                                     GObject *recall_container);
```

Removes a recall container.

Parameters

channel	an AgsChannel	
recall_container	the AgsRecallContainer	

Since: 3.0.0

ags_channel_get_play ()

```
GList~*
ags_channel_get_play (AgsChannel *channel);
```

Get play.

Parameters

channel	the AgsChannel	

Returns

the GList containig AgsRecall.

[element-type AgsAudio.Recall][transfer full]

Since: 3.1.0

ags_channel_set_play ()

```
void
ags_channel_set_play (AgsChannel *channel,
                      GList *play);
```

Set play by replacing existing.

Parameters

channel	the AgsChannel	
play	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer full]*

Since: 3.1.0

ags_channel_get_recall ()

```
GList~*
ags_channel_get_recall (AgsChannel *channel);
```

Get recall.

Parameters

channel	the AgsChannel	

Returns

the GList containig AgsRecall.

[element-type AgsAudio.Recall][transfer full]

Since: 3.1.0

ags_channel_set_recall ()

```
void
ags_channel_set_recall (AgsChannel *channel,
                        GList *recall);
```

Set recall by replacing existing.

Parameters

channel	the AgsChannel	
recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer full]*

Since: 3.1.0

ags_channel_add_recall ()

```
void
ags_channel_add_recall (AgsChannel *channel,
                        GObject *recall,
                        gboolean play_context);
```

Adds a recall.

Parameters

channel	an AgsChannel	
recall	the AgsRecall	
play_context	TRUE if play context, else if FALSE recall context	

Since: 3.0.0

ags_channel_insert_recall ()

```
void
ags_channel_insert_recall (AgsChannel *channel,
                           GObject *recall,
                           gboolean play_context,
                           gint position);
```

Insert *recall* at *position* in *channel* 's *play_context* .

Parameters

channel	an AgsChannel	
recall	the AgsRecall	
play_context	TRUE if play context, else if FALSE recall context	
position	the position	

Since: 3.3.0

ags_channel_remove_recall ()

```
void
ags_channel_remove_recall (AgsChannel *channel,
                           GObject *recall,
                           gboolean play_context);
```

Removes a recall.

Parameters

channel	an AgsChannel	
recall	the AgsRecall	
play_context	TRUE if play context, else if FALSE recall context	

Since: 3.0.0

ags_channel_duplicate_recall ()

```
void
ags_channel_duplicate_recall (AgsChannel *channel,
                              AgsRecallID *recall_id);
```

Duplicate AgsRecall template and assign *recall_id* to it.

Parameters

channel	the AgsChannel	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_channel_resolve_recall ()

```
void
ags_channel_resolve_recall (AgsChannel *channel,
                            AgsRecallID *recall_id);
```

Resolve step of initialization.

Parameters

channel	an AgsChannel	
recall_id	appropriate AgsRecallID	

Since: 3.0.0

ags_channel_init_recall ()

```
void
ags_channel_init_recall (AgsChannel *channel,
                         AgsRecallID *recall_id,
                         guint staging_flags);
```

Prepare AgsRecall objects and invoke "run-init-pre", "run-init-inter" or "run-init-post" as specified by *staging_flags* .

Parameters

channel	the AgsChannel	
recall_id	the AgsRecallID	
staging_flags	the stages to invoke	

Since: 3.0.0

ags_channel_play_recall ()

```
void
ags_channel_play_recall (AgsChannel *channel,
                         AgsRecallID *recall_id,
                         guint staging_flags);
```

Run the specified steps by *recall_id* of *channel* .

Parameters

channel	the AgsChannel	
recall_id	the AgsRecallID	
staging_flags	the stages to invoke	

Since: 3.0.0

ags_channel_done_recall ()

```
void
ags_channel_done_recall (AgsChannel *channel,
                         AgsRecallID *recall_id);
```

Done processing specified by *recall_id* .

Parameters

channel	the AgsChannel	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_channel_cancel_recall ()

```
void
ags_channel_cancel_recall (AgsChannel *channel,
                           AgsRecallID *recall_id);
```

Cancel processing specified by *recall_id* .

Parameters

channel	the AgsChannel	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_channel_cleanup_recall ()

```
void
ags_channel_cleanup_recall (AgsChannel *channel,
                            AgsRecallID *recall_id);
```

Cleanup processing specified by *recall_id* .

Parameters

channel	the AgsChannel	
recall_id	the AgsRecallID	

Since: 3.0.0

ags_channel_start ()

```
GList~*
ags_channel_start (AgsChannel *channel,
                   gint sound_scope);
```

Start *channel* 's *sound_scope* to do playback.

Parameters

channel	the AgsChannel	
sound_scope	the sound scope	

Returns

the GList containing AgsRecallID.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.0.0

ags_channel_stop ()

```
void
ags_channel_stop (AgsChannel *channel,
                  GList *recall_id,
                  gint sound_scope);
```

Stop *channel* 's *sound_scope* playback specified by *recall_id* .

Parameters

channel	the AgsChannel	
recall_id	the GList containing AgsRecallID.	*[element-type AgsAudio.RecallID][transfer none]*
sound_scope	the sound scope	

Since: 3.0.0

ags_channel_check_scope ()

```
GList~*
ags_channel_check_scope (AgsChannel *channel,
                         gint sound_scope);
```

Check *channel* 's *sound_scope* .

Parameters

channel	the AgsChannel	
sound_scope	the sound scope	

Returns

the GList containing AgsRecallID or NULL if not playing.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.0.0

ags_channel_collect_all_channel_ports ()

```
GList~*
ags_channel_collect_all_channel_ports (AgsChannel *channel);
```

Retrieve all ports of AgsChannel.

Parameters

channel	the AgsChannel	

Returns

a new GList containing AgsPort.

[element-type AgsAudio.Port][transfer full]

Since: 3.0.0

ags_channel_collect_all_channel_ports_by_specifier_and_context ()

```
GList~*
ags_channel_collect_all_channel_ports_by_specifier_and_context
                                (AgsChannel *channel,
                                 gchar *specifier,
                                 gboolean play_context);
```

Retrieve specified port of AgsChannel

Parameters

channel	an AgsChannel	
specifier	the port's name	
play_context	either TRUE for play or FALSE for recall	

Returns

a GList of AgsPort if found, otherwise NULL.

[element-type AgsAudio.Port][transfer full]

Since: 3.0.0

ags_channel_get_level ()

```
AgsChannel~*
ags_channel_get_level (AgsChannel *channel);
```

Get level.

Parameters

channel	the AgsChannel	

Returns

the level of `channel` as AgsChannel or NULL if no parent recycling.

[transfer full]

Since: 3.0.0

ags_channel_recursive_set_property ()

```
void
ags_channel_recursive_set_property (AgsChannel *channel,
                                    gint n_params,
                                    gchar **parameter_name,
                                    GValue *value);
```

Recursive set property for AgsChannel.

Parameters

channel	the AgsChannel	
n_params	the count of paramter name and value pairs	
parameter_name	a string vector containing parameter names	
value	the value array	

Since: 3.0.0

ags_channel_recursive_run_stage ()

```
void
ags_channel_recursive_run_stage (AgsChannel *channel,
                                 gint sound_scope,
                                 guint staging_flags);
```

Recursive run stage specified by *staging_flags* for matching *sound_scope* .

Parameters

channel	the AgsChannel	
sound_scope	the sound scope	
staging_flags	the staging flags	

Since: 3.0.0

ags_channel_new ()

```
AgsChannel~*
ags_channel_new (GObject *audio);
```

Creates a new instance of AgsChannel, linking tree to *audio* .

Parameters

audio	the AgsAudio	

Returns

a new AgsChannel

Since: 3.0.0

AGS_CHANNEL()

```
#define AGS_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_CHANNEL ↩
    , AgsChannel))
```

AGS_CHANNEL_CLASS()

```
#define AGS_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_CHANNEL, ↩
    AgsChannelClass))
```

AGS_CHANNEL_GET_CLASS()

```
#define AGS_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_CHANNEL, ↩
    AgsChannelClass))
```

AGS_IS_CHANNEL()

```
#define AGS_IS_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_CHANNEL ↩
    ))
```

AGS_IS_CHANNEL_CLASS()

```
#define AGS_IS_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_CHANNEL ↩
    ))
```

ags_channel_flags_get_type ()

```
GType
ags_channel_flags_get_type ();
```

ags_channel_get_type ()

```
GType
ags_channel_get_type ();
```

Types and Values

AGS_CHANNEL_MINIMUM_OCTAVE

```
#define AGS_CHANNEL_MINIMUM_OCTAVE (0)
```

AGS_CHANNEL_MAXIMUM_OCTAVE

```
#define AGS_CHANNEL_MAXIMUM_OCTAVE (10)
```

AGS_CHANNEL_DEFAULT_OCTAVE

```
#define AGS_CHANNEL_DEFAULT_OCTAVE (AGS_CHANNEL_MINIMUM_OCTAVE)
```

AGS_CHANNEL_MINIMUM_OCTAVE_SEMITONE

```
#define AGS_CHANNEL_MINIMUM_OCTAVE_SEMITONE (0)
```

AGS_CHANNEL_MAXIMUM_OCTAVE_SEMITONE

```
#define AGS_CHANNEL_MAXIMUM_OCTAVE_SEMITONE (12)
```

AGS_CHANNEL_DEFAULT_OCTAVE_SEMITONE

```
#define AGS_CHANNEL_DEFAULT_OCTAVE_SEMITONE (AGS_CHANNEL_MINIMUM_OCTAVE_SEMITONE)
```

AGS_CHANNEL_MINIMUM_SEMITONE

```
#define AGS_CHANNEL_MINIMUM_SEMITONE (0)
```

AGS_CHANNEL_MAXIMUM_SEMITONE

```
#define AGS_CHANNEL_MAXIMUM_SEMITONE (128)
```

AGS_CHANNEL_DEFAULT_SEMITONE

```
#define AGS_CHANNEL_DEFAULT_SEMITONE (AGS_CHANNEL_MINIMUM_SEMITONE)
```

AGS_CHANNEL_MINIMUM_NOTE_FREQUENCY

```
#define AGS_CHANNEL_MINIMUM_NOTE_FREQUENCY (440.0 * exp((-69.0 / 12.0) * log(2.0)))
```

AGS_CHANNEL_MAXIMUM_NOTE_FREQUENCY

```
#define AGS_CHANNEL_MAXIMUM_NOTE_FREQUENCY (440.0 * exp((58.0 / 12.0) * log(2.0)))
```

AGS_CHANNEL_DEFAULT_NOTE_FREQUENCY

```
#define AGS_CHANNEL_DEFAULT_NOTE_FREQUENCY (AGS_CHANNEL_MINIMUM_NOTE_FREQUENCY)
```

AGS_CHANNEL_MINIMUM_MIDI_NOTE

```
#define AGS_CHANNEL_MINIMUM_MIDI_NOTE (0)
```

AGS_CHANNEL_MAXIMUM_MIDI_NOTE

```
#define AGS_CHANNEL_MAXIMUM_MIDI_NOTE (127)
```

AGS_CHANNEL_DEFAULT_MIDI_NOTE

```
#define AGS_CHANNEL_DEFAULT_MIDI_NOTE (AGS_CHANNEL_MINIMUM_MIDI_NOTE)
```

enum AgsChannelFlags

Enum values to control the behavior or indicate internal state of AgsChannel by enable/disable as flags.

Members

AGS_CHANNEL_BYPASS	don't apply any data

AGS_CHANNEL_ERROR

```
#define AGS_CHANNEL_ERROR (ags_channel_error_quark())
```

enum AgsChannelError

Members

AGS_CHANNEL_ERROR_LOOP_IN_LINK	

AGS_TYPE_CHANNEL

```
#define AGS_TYPE_CHANNEL                (ags_channel_get_type())
```

struct AgsChannel

```
struct AgsChannel;
```

struct AgsChannelClass

```
struct AgsChannelClass {
  GObjectClass gobject;

  void (*recycling_changed)(AgsChannel *channel,
          AgsRecycling *old_start_region, AgsRecycling *old_end_region,
          AgsRecycling *new_start_region, AgsRecycling *new_end_region,
          AgsRecycling *old_start_changed_region, AgsRecycling *old_end_changed_region,
          AgsRecycling *new_start_changed_region, AgsRecycling *new_end_changed_region);

  void (*duplicate_recall)(AgsChannel *channel,
         AgsRecallID *recall_id);
  void (*resolve_recall)(AgsChannel *channel,
       AgsRecallID *recall_id);
```

```
  void (*init_recall)(AgsChannel *channel,
          AgsRecallID *recall_id, guint staging_flags);
  void (*play_recall)(AgsChannel *channel,
          AgsRecallID *recall_id, guint staging_flags);

  void (*done_recall)(AgsChannel *channel,
          AgsRecallID *recall_id);
  void (*cancel_recall)(AgsChannel *channel,
      AgsRecallID *recall_id);

  void (*cleanup_recall)(AgsChannel *channel,
       AgsRecallID *recall_id);

  GList* (*start)(AgsChannel *channel,
      gint sound_scope);
  void (*stop)(AgsChannel *channel,
         GList *recall_id, gint sound_scope);

  GList* (*check_scope)(AgsChannel *channel, gint sound_scope);

  void (*recursive_run_stage)(AgsChannel *channel,
            gint sound_scope, guint staging_flags);
};
```

Property Details

The "absolute-key" property

```
  "absolute-key"              int
```

The nth absolute key.

Owner: AgsChannel

Flags: Read / Write

Allowed values: [0,128]

Default value: 0

Since: 3.0.0

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio aligning channels.

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "audio-channel" property

```
  "audio-channel"             guint
```

The nth audio channel.

Owner: AgsChannel

Flags: Read

Default value: 0

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size.

Owner: AgsChannel

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "first-recycling" property

```
"first-recycling"            AgsRecycling~*
```

The containing AgsRecycling it takes it AgsAudioSignal from.

Owner: AgsChannel

Flags: Read

Since: 3.0.0

The "format" property

```
"format"                     guint
```

The format.

Owner: AgsChannel

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "input-soundcard" property

```
"input-soundcard"            GObject~*
```

The assigned input AgsSoundcard.

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "input-soundcard-channel" property

```
"input-soundcard-channel"  int
```

The input soundcard channel.

Owner: AgsChannel

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "key" property

```
"key"                      guint
```

The nth key.

Owner: AgsChannel

Flags: Read / Write

Allowed values: <= 12

Default value: 0

Since: 3.0.0

The "last-recycling" property

```
"last-recycling"           AgsRecycling~*
```

The containing AgsRecycling it takes it AgsAudioSignal from.

Owner: AgsChannel

Flags: Read

Since: 3.0.0

The "line" property

```
"line"                     guint
```

The nth line.

Owner: AgsChannel

Flags: Read

Default value: 0

Since: 3.0.0

The "link" property

```
"link"                    AgsChannel~*
```

The assigned link as AgsChannel.

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "midi-note" property

```
"midi-note"               guint
```

The nth midi note.

Owner: AgsChannel

Flags: Read / Write

Allowed values: <= 127

Default value: 0

Since: 3.0.0

The "next" property

```
"next"                    AgsChannel~*
```

The assigned next AgsChannel.

Owner: AgsChannel

Flags: Read

Since: 3.0.0

The "next-pad" property

```
"next-pad"                AgsChannel~*
```

The assigned next pad AgsChannel.

Owner: AgsChannel

Flags: Read

Since: 3.0.0

The "note-frequency" property

```
"note-frequency"          double
```

The note frequency.

Owner: AgsChannel

Flags: Read / Write

Allowed values: [8.1758,12543.9]

Default value: 8.1758

Since: 3.0.0

The "note-key" property

```
"note-key"                  char~*
```

The assigned note key representing this channel.

Owner: AgsChannel

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "octave" property

```
"octave"                    int
```

The nth octave.

Owner: AgsChannel

Flags: Read / Write

Allowed values: [0,10]

Default value: 0

Since: 3.0.0

The "output-soundcard" property

```
"output-soundcard"          GObject~*
```

The assigned output AgsSoundcard.

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "output-soundcard-channel" property

```
"output-soundcard-channel" int
```

The output soundcard channel.

Owner: AgsChannel

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "pad" property

```
"pad"                    guint
```

The nth pad.

Owner: AgsChannel

Flags: Read

Default value: 0

Since: 3.0.0

The "pattern" property

```
"pattern"                gpointer
```

The containing AgsPattern.

[transfer full]

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "play" property

```
"play"                   gpointer
```

The containing AgsRecall in play-context.

[transfer full]

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "playback" property

```
"playback"               AgsPlayback~*
```

The assigned AgsPlayback.

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "prev" property

```
"prev"                   AgsChannel~*
```

The assigned prev AgsChannel.

Owner: AgsChannel

Flags: Read

Since: 3.0.0

The "prev-pad" property

```
"prev-pad"                  AgsChannel~*
```

The assigned prev pad AgsChannel.

Owner: AgsChannel

Flags: Read

Since: 3.0.0

The "recall" property

```
"recall"                    gpointer
```

The containing AgsRecall in recall-context.

[transfer full]

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "recall-container" property

```
"recall-container"          gpointer
```

The containing AgsRecallContainer.

[transfer full]

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "recall-id" property

```
"recall-id"                 gpointer
```

The assigned AgsRecallID.

[transfer full]

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "recycling-context" property

```
"recycling-context"         gpointer
```

The containing AgsRecyclingContext.

[transfer full]

Owner: AgsChannel

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
  "samplerate"                guint
```

The samplerate.

Owner: AgsChannel

Flags: Read / Write

Default value: 0

Since: 3.0.0

Signal Details

The "cancel-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               gpointer    user_data)
```

The ::cancel-recall signal is invoked during termination of playback.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "check-scope" signal

```
gpointer
user_function (AgsChannel *channel,
               int         sound_scope,
               gpointer    user_data)
```

The ::check-scope signal gives you control of checking scope.

Parameters

channel	the AgsChannel	
sound_scope	the sound scope	
user_data	user data set when the signal handler was connected.	

Returns

the GList containing AgsRecallID

Flags: Run Last

Since: 3.0.0

The "cleanup-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               gpointer    user_data)
```

The ::cleanup-recall signal is invoked during termination of playback.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "done-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               gpointer    user_data)
```

The ::done-recall signal is invoked during termination of playback.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "duplicate-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               gpointer    user_data)
```

The ::duplicate-recall signal is invoked during playback initialization.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "init-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               guint       staging_flags,
               gpointer    user_data)
```

The ::init-recall signal is invoked during playback initialization.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
staging_flags	the staging flags	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "play-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               guint       staging_flags,
               gpointer    user_data)
```

The ::play-recall signal is invoked during playback run.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
staging_flags	the staging flags	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "recursive-run-stage" signal

```
void
user_function (AgsChannel *channel,
               int         sound_scope,
               guint       staging_flags,
               gpointer    user_data)
```

The ::recursive-run-stage signal gives you control of checking scope.

Parameters

channel	the AgsChannel	
sound_scope	the sound scope	
staging_flags	the staging flags	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "recycling-changed" signal

```
void
user_function (AgsChannel *channel,
               GObject    *old_start_region,
               GObject    *old_end_region,
               GObject    *new_start_region,
               GObject    *new_end_region,
               GObject    *old_start_changed_region,
               GObject    *old_end_changed_region,
               GObject    *new_start_changed_region,
               GObject    *new_end_changed_region,
               gpointer    user_data)
```

The ::recycling-changed signal is invoked to notify modified recycling tree.

Parameters

channel	the AgsChannel recycling changed	
old_start_region	first AgsRecycling	
old_end_region	last AgsRecycling	
new_start_region	new first AgsRecycling	
new_end_region	new last AgsRecycling	
old_start_changed_region	modified link AgsRecycling start	
old_end_changed_region	modified link AgsRecyclig end	
new_start_changed_region	replacing link AgsRecycling start	
new_end_changed_region	replacing link AgsRecycling end	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "resolve-recall" signal

```
void
user_function (AgsChannel *channel,
               GObject    *recall_id,
               gpointer    user_data)
```

The ::resolve-recall signal is invoked during playback initialization.

Parameters

channel	the AgsChannel	
recall_id	the appropriate AgsRecallID	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "start" signal

```
gpointer
user_function (AgsChannel *channel,
               int         sound_scope,
               gpointer    user_data)
```

The ::start signal is invoked as playback starts.

Parameters

channel	the AgsChannel	
sound_scope	the sound scope	
user_data	user data set when the signal handler was connected.	

Returns

the GList containing AgsRecallID.

[type GLib.List][element-type AgsAudio.RecallID][transfer full]

Flags: Run Last

Since: 3.0.0

The "stop" signal

```
void
user_function (AgsChannel *channel,
               gpointer    recall_id,
               int         sound_scope,
               gpointer    user_data)
```

The ::stop signal is invoked as playback stops.

Parameters

channel	the AgsChannel	
recall_id	the GList containing AgsRecallID	
sound_scope	the sound scope	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

9.5 AgsOutput

AgsOutput — Output of AgsAudio

Stability Level

Stable, unless otherwise indicated

Functions

AgsRecycling *	ags_output_find_first_input_recycling ()
AgsRecycling *	ags_output_find_last_input_recycling ()
AgsOutput *	ags_output_new ()
#define	AGS_IS_OUTPUT()
#define	AGS_IS_OUTPUT_CLASS()
#define	AGS_OUTPUT()
#define	AGS_OUTPUT_CLASS()
#define	AGS_OUTPUT_GET_CLASS()
GType	ags_output_get_type ()

Types and Values

#define	AGS_TYPE_OUTPUT
struct	AgsOutput
struct	AgsOutputClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsChannel
        &#x2570;&#x2500;&#x2500; AgsOutput
```

Implemented Interfaces

AgsOutput implements AgsConnectable.

Includes

```
#include <ags/audio/ags_output.h>
```

Description

AgsOutput represents a output channel of AgsAudio.

Functions

ags_output_find_first_input_recycling ()

```
AgsRecycling~*
ags_output_find_first_input_recycling (AgsOutput *output);
```

Retrieve first input recycling.

Parameters

output	an AgsOutput

Returns

the first AgsRecycling of AgsAudio.

[transfer full]

Since: 3.0.0

ags_output_find_last_input_recycling ()

```
AgsRecycling~*
ags_output_find_last_input_recycling (AgsOutput *output);
```

Retrieve last input recycling.

Parameters

output	an AgsOutput

Returns

the last AgsRecycling of AgsAudio.

[transfer full]

Since: 3.0.0

ags_output_new ()

```
AgsOutput~*
ags_output_new (GObject *audio);
```

Creates an AgsOutput, assigned to *audio* .

Parameters

audio	the AgsAudio

Returns

a new AgsOutput

Since: 3.0.0

AGS_IS_OUTPUT()

```
#define AGS_IS_OUTPUT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_OUTPUT))
```

AGS_IS_OUTPUT_CLASS()

```
#define AGS_IS_OUTPUT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_OUTPUT))
```

AGS_OUTPUT()

```
#define AGS_OUTPUT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_OUTPUT,  ↩
    AgsOutput))
```

AGS_OUTPUT_CLASS()

```
#define AGS_OUTPUT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_OUTPUT,  ↩
    AgsOutputClass))
```

AGS_OUTPUT_GET_CLASS()

```
#define AGS_OUTPUT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj), AGS_TYPE_OUTPUT,  ↩
    AgsOutputClass))
```

ags_output_get_type ()

```
GType
ags_output_get_type ();
```

Types and Values

AGS_TYPE_OUTPUT

```
#define AGS_TYPE_OUTPUT                 (ags_output_get_type())
```

struct AgsOutput

```
struct AgsOutput;
```

struct AgsOutputClass

```
struct AgsOutputClass {
  AgsChannelClass channel;
};
```

9.6 AgsInput

AgsInput — Input of AgsAudio

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_input_is_active ()
AgsInput *	ags_input_next_active ()
void	ags_input_add_synth_generator ()
void	ags_input_remove_synth_generator ()
gboolean	ags_input_open_file ()
AgsInput *	ags_input_new ()
#define	AGS_INPUT()
#define	AGS_INPUT_CLASS()
#define	AGS_INPUT_GET_CLASS()
#define	AGS_IS_INPUT()
#define	AGS_IS_INPUT_CLASS()
GType	ags_input_get_type ()

Properties

AgsFileLink *	file-link	Read / Write
gpointer	synth-generator	Read / Write

Types and Values

#define	AGS_INPUT_SYNTH_BASE_NOTE
#define	AGS_TYPE_INPUT
struct	AgsInput
struct	AgsInputClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsChannel
    &#x2570;&#x2500;&#x2500; AgsInput
```

Implemented Interfaces

AgsInput implements AgsConnectable.

Includes

```
#include <ags/audio/ags_input.h>
```

Description

AgsInput represents an input channel of AgsAudio.

Functions

ags_input_is_active ()

```
gboolean
ags_input_is_active (AgsInput *input,
                     GObject *recycling_context);
```

Check if *input* is active and needs processing.

Parameters

input	the AgsInput	
recycling_context	the AgsRecyclingContext to check	

Returns

TRUE if has a need to be processed, else FALSE

Since: 3.0.0

ags_input_next_active ()

```
AgsInput~*
ags_input_next_active (AgsInput *input,
                       GObject *recycling_context);
```

Find next AgsInput needed to be processed.

Parameters

input	the AgsInput	
recycling_context	the AgsRecyclingContext to check	

Returns

next active AgsInput, else NULL if non available.

[transfer full]

Since: 3.0.0

ags_input_add_synth_generator ()

```
void
ags_input_add_synth_generator (AgsInput *input,
                               GObject *synth_generator);
```

Add *synth_generator* to *input* .

Parameters

input	the AgsInput	
synth_generator	the AgsSynthGenerator	

Since: 3.0.0

ags_input_remove_synth_generator ()

```
void
ags_input_remove_synth_generator (AgsInput *input,
                                  GObject *synth_generator);
```

Remove *synth_generator* from *input* .

Parameters

input	the AgsInput	
synth_generator	the AgsSynthGenerator	

Since: 3.0.0

ags_input_open_file ()

```
gboolean
ags_input_open_file (AgsInput *input,
                     gchar *filename,
                     gchar *preset,
                     gchar *instrument,
                     gchar *sample,
                     guint audio_channel);
```

Open *filename* and assign *audio_channel* 's data as AGS_AUDIO_SIGNAL_TEMPLATE audio signal to *input* 's own recycling.

Parameters

input	the AgsInput	
filename	the filename as string	
preset	the preset to open	
instrument	the instrument to open	
sample	the sample to open	
audio_channel	the audio channel to read	

Returns

TRUE if open was successful, else FALSE

Since: 3.0.0

ags_input_new ()

```
AgsInput~*
ags_input_new (GObject *audio);
```

Creates a AgsInput, linking tree to *audio* .

Parameters

audio	the AgsAudio

Returns

a new AgsInput

Since: 3.0.0

AGS_INPUT()

```
#define AGS_INPUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_INPUT,  ↩
    AgsInput))
```

AGS_INPUT_CLASS()

```
#define AGS_INPUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_INPUT,  ↩
    AgsInputClass))
```

AGS_INPUT_GET_CLASS()

```
#define AGS_INPUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_INPUT,  ↩
    AgsInputClass))
```

AGS_IS_INPUT()

```
#define AGS_IS_INPUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_INPUT))
```

AGS_IS_INPUT_CLASS()

```
#define AGS_IS_INPUT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_INPUT))
```

ags_input_get_type ()

```
GType
ags_input_get_type ();
```

Types and Values

AGS_INPUT_SYNTH_BASE_NOTE

```
#define AGS_INPUT_SYNTH_BASE_NOTE (-48.0)
```

AGS_TYPE_INPUT

```
#define AGS_TYPE_INPUT                (ags_input_get_type())
```

struct AgsInput

```
struct AgsInput;
```

struct AgsInputClass

```
struct AgsInputClass {
  AgsChannelClass channel;
};
```

Property Details

The "file-link" property

```
  "file-link"                 AgsFileLink~*
```

The file containing audio data.

Owner: AgsInput

Flags: Read / Write

Since: 3.0.0

The "synth-generator" property

```
  "synth-generator"           gpointer
```

The synth generators assigned with this input.

[transfer full]

Owner: AgsInput

Flags: Read / Write

Since: 3.0.0

9.7 AgsRecycling

AgsRecycling — Recycling container of audio signals

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RECYCLING_GET_OBJ_MUTEX()
GRecMutex *	ags_recycling_get_obj_mutex ()
gboolean	ags_recycling_test_flags ()
void	ags_recycling_set_flags ()
void	ags_recycling_unset_flags ()
GObject *	ags_recycling_get_channel ()
void	ags_recycling_set_channel ()
AgsRecycling *	ags_recycling_next ()
AgsRecycling *	ags_recycling_prev ()
GObject *	ags_recycling_get_output_soundcard ()
void	ags_recycling_set_output_soundcard ()
GObject *	ags_recycling_get_input_soundcard ()
void	ags_recycling_set_input_soundcard ()
guint	ags_recycling_get_samplerate ()
void	ags_recycling_set_samplerate ()
guint	ags_recycling_get_buffer_size ()
void	ags_recycling_set_buffer_size ()
guint	ags_recycling_get_format ()
void	ags_recycling_set_format ()
GList *	ags_recycling_get_audio_signal ()
void	ags_recycling_set_audio_signal ()
void	ags_recycling_add_audio_signal ()
void	ags_recycling_remove_audio_signal ()
void	ags_recycling_data_request ()
gint	ags_recycling_position ()
AgsRecycling *	ags_recycling_find_next_channel ()
gboolean	ags_recycling_is_active ()
void	ags_recycling_create_audio_signal_with_defaults ()
void	ags_recycling_create_audio_signal_with_frame_count ()
AgsRecycling *	ags_recycling_new ()
#define	AGS_IS_RECYCLING()
#define	AGS_IS_RECYCLING_CLASS()
#define	AGS_RECYCLING()
#define	AGS_RECYCLING_CLASS()
#define	AGS_RECYCLING_GET_CLASS()
GType	ags_recycling_flags_get_type ()
GType	ags_recycling_get_type ()

Properties

gpointer	audio-signal	Read / Write
guint	buffer-size	Read / Write
AgsChannel *	channel	Read / Write
guint	format	Read / Write
GObject *	input-soundcard	Read / Write
int	input-soundcard-channel	Read / Write
AgsRecycling *	next	Read / Write
GObject *	output-soundcard	Read / Write

int	output-soundcard-channel	Read / Write
AgsRecycling *	parent	Read / Write
AgsRecycling *	prev	Read / Write
guint	samplerate	Read / Write

Signals

void	add-audio-signal	Run Last
void	data-request	Run Last
void	remove-audio-signal	Run Last

Types and Values

enum	AgsRecyclingFlags
#define	AGS_TYPE_RECYCLING
struct	AgsRecycling
struct	AgsRecyclingClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsRecyclingFlags
GObject
&#x2570;&#x2500;&#x2500; AgsRecycling
```

Implemented Interfaces

AgsRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recycling.h>
```

Description

AgsRecycling forms the nested tree of AgsAudio and AgsChannel. The recycling tree is the actual audio processing tree, because it hosts AgsAudioSignal.

Every recycling node introduces a new AgsAudioSignal for specific AgsSoundScope specified by assigned AgsRecallID.

All audio processing happens on AgsAudioSignal and is added to AgsRecycling by ags_recycling_add_audio_signal(). While you call it, the function takes the audio signal with AGS_AUDIO_SIGNAL_TEMPLATE flag set and creates a copy AgsAudioSignal with the very same audio data of `stream` property.

The actual audio processing functionality is provided by AgsRecall implementations. Its entry point is AgsAudio or AgsChannel.

AgsRecall objectss except templates are assigned with a AgsRecallID with a relation to AgsRecyclingContext.

Functions

AGS_RECYCLING_GET_OBJ_MUTEX()

```
#define AGS_RECYCLING_GET_OBJ_MUTEX(obj) (&(((AgsRecycling *) obj)->obj_mutex))
```

ags_recycling_get_obj_mutex ()

```
GRecMutex~*
ags_recycling_get_obj_mutex (AgsRecycling *recycling);
```

Get object mutex.

Parameters

recycling	the AgsRecycling

Returns

the GRecMutex to lock *recycling*

Since: 3.1.0

ags_recycling_test_flags ()

```
gboolean
ags_recycling_test_flags (AgsRecycling *recycling,
                          guint flags);
```

Test *flags* to be set on *recycling* .

Parameters

recycling	the AgsRecycling
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recycling_set_flags ()

```
void
ags_recycling_set_flags (AgsRecycling *recycling,
                         guint flags);
```

Enable a feature of *recycling* .

Parameters

recycling	the AgsRecycling
flags	see AgsRecyclingFlags

Since: 3.0.0

ags_recycling_unset_flags ()

```
void
ags_recycling_unset_flags (AgsRecycling *recycling,
                           guint flags);
```

Disable a feature of *recycling* .

Parameters

recycling	the AgsRecycling	
flags	see AgsRecyclingFlags	

Since: 3.0.0

ags_recycling_get_channel ()

```
GObject~*
ags_recycling_get_channel (AgsRecycling *recycling);
```

Get channel.

Parameters

recycling	the AgsRecycling	

Returns

the AgsChannel.

[transfer full]

Since: 3.1.0

ags_recycling_set_channel ()

```
void
ags_recycling_set_channel (AgsRecycling *recycling,
                           GObject *channel);
```

Set channel.

Parameters

recycling	the AgsRecycling	
channel	the AgsChannel	

Since: 3.1.0

ags_recycling_next ()

```
AgsRecycling~*
ags_recycling_next (AgsRecycling *recycling);
```

Iterate *recycling* .

Parameters

recycling	the AgsRecycling

Returns

the next of AgsRecycling if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_recycling_prev ()

```
AgsRecycling~*
ags_recycling_prev (AgsRecycling *recycling);
```

Iterate *recycling* .

Parameters

recycling	the AgsRecycling

Returns

the prev of AgsRecycling if available, otherwise NULL.

[transfer full]

Since: 3.0.0

ags_recycling_get_output_soundcard ()

```
GObject~*
ags_recycling_get_output_soundcard (AgsRecycling *recycling);
```

Get the output soundcard object of *recycling* .

Parameters

recycling	the AgsRecycling

Returns

the output soundcard.

[transfer full]

Since: 3.1.0

ags_recycling_set_output_soundcard ()

```
void
ags_recycling_set_output_soundcard (AgsRecycling *recycling,
                                    GObject *output_soundcard);
```

Set the output soundcard object of *recycling* .

Parameters

recycling	the AgsRecycling	
output_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_recycling_get_input_soundcard ()

```
GObject~*
ags_recycling_get_input_soundcard (AgsRecycling *recycling);
```

Get the input soundcard object of *recycling* .

Parameters

recycling	the AgsRecycling	

Returns

the input soundcard.

[transfer full]

Since: 3.1.0

ags_recycling_set_input_soundcard ()

```
void
ags_recycling_set_input_soundcard (AgsRecycling *recycling,
                                   GObject *input_soundcard);
```

Set the input soundcard object of *recycling* .

Parameters

recycling	an AgsRecycling	
input_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_recycling_get_samplerate ()

```
guint
ags_recycling_get_samplerate (AgsRecycling *recycling);
```

Gets samplerate.

Parameters

recycling	the AgsRecycling

Returns

the samplerate

Since: 3.1.0

ags_recycling_set_samplerate ()

```
void
ags_recycling_set_samplerate (AgsRecycling *recycling,
                              guint samplerate);
```

Sets samplerate.

Parameters

recycling	the AgsRecycling
samplerate	the samplerate

Since: 3.0.0

ags_recycling_get_buffer_size ()

```
guint
ags_recycling_get_buffer_size (AgsRecycling *recycling);
```

Gets buffer size.

Parameters

recycling	the AgsRecycling

Returns

the buffer size

Since: 3.1.0

ags_recycling_set_buffer_size ()

```
void
ags_recycling_set_buffer_size (AgsRecycling *recycling,
                               guint buffer_size);
```

Set buffer size.

Parameters

recycling	the AgsRecycling	
buffer_size	the buffer size	

Since: 3.0.0

ags_recycling_get_format ()

```
guint
ags_recycling_get_format (AgsRecycling *recycling);
```

Gets format.

Parameters

recycling	the AgsRecycling	

Returns

the format

Since: 3.1.0

ags_recycling_set_format ()

```
void
ags_recycling_set_format (AgsRecycling *recycling,
                          guint format);
```

Set format.

Parameters

recycling	the AgsRecycling	
format	the format	

Since: 3.0.0

ags_recycling_get_audio_signal ()

```
GList~*
ags_recycling_get_audio_signal (AgsRecycling *recycling);
```

Get recall id.

Parameters

recycling	the AgsRecycling	

Returns

the GList containig AgsAudioSignal.

[element-type AgsAudio.AudioSignal][transfer full]

Since: 3.1.0

ags_recycling_set_audio_signal ()

```
void
ags_recycling_set_audio_signal (AgsRecycling *recycling,
                                GList *audio_signal);
```

Set recall id by replacing existing.

Parameters

recycling	the AgsRecycling	
audio_signal	the GList containing AgsAudioSignal.	*[element-type AgsAudio.AudioSignal][transfer full]*

Since: 3.1.0

ags_recycling_add_audio_signal ()

```
void
ags_recycling_add_audio_signal (AgsRecycling *recycling,
                                AgsAudioSignal *audio_signal);
```

Add `audio_signal` to `recycling` .

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal to add	

Since: 3.0.0

ags_recycling_remove_audio_signal ()

```
void
ags_recycling_remove_audio_signal (AgsRecycling *recycling,
                                   AgsAudioSignal *audio_signal);
```

Remove `audio_signal` of `recycling` .

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal to remove	

Since: 3.0.0

ags_recycling_data_request ()

```
void
ags_recycling_data_request (AgsRecycling *recycling,
                            AgsAudioSignal *audio_signal);
```

Request data of *audio_signal* .

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal	

Since: 3.0.0

ags_recycling_position ()

```
gint
ags_recycling_position (AgsRecycling *start_region,
                        AgsRecycling *end_region,
                        AgsRecycling *recycling);
```

Retrieve position of recycling.

Parameters

start_region	boundary start	
end_region	boundary end	
recycling	matching recycling	

Returns

position within boundary.

Since: 3.0.0

ags_recycling_find_next_channel ()

```
AgsRecycling~*
ags_recycling_find_next_channel (AgsRecycling *start_region,
                                 AgsRecycling *end_region,
                                 GObject *prev_channel);
```

Retrieve next recycling with different channel.

Parameters

start_region	boundary start	
end_region	boundary end	
prev_channel	previous channel	

Returns

Matching recycling.

[transfer full]

Since: 3.0.0

ags_recycling_is_active ()

```
gboolean
ags_recycling_is_active (AgsRecycling *start_region,
                         AgsRecycling *end_region,
                         GObject *recall_id);
```

Check if is active.

Parameters

start_region	boundary start	
end_region	boundary end	
recall_id	the AgsRecallID	

Returns

TRUE if related audio signal to recall id is available, otherwise FALSE

Since: 3.0.0

ags_recycling_create_audio_signal_with_defaults ()

```
void
ags_recycling_create_audio_signal_with_defaults
                                (AgsRecycling *recycling,
                                 AgsAudioSignal *audio_signal,
                                 gdouble delay,
                                 guint attack);
```

Create audio signal with defaults.

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal to apply defaults	
delay	the delay	
attack	the attack	

Since: 3.0.0

ags_recycling_create_audio_signal_with_frame_count ()

```
void
ags_recycling_create_audio_signal_with_frame_count
                                (AgsRecycling *recycling,
                                 AgsAudioSignal *audio_signal,
                                 guint frame_count,
                                 gdouble delay,
                                 guint attack);
```

Create audio signal with frame count.

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal to apply defaults	
frame_count	the audio data size	
delay	the delay	
attack	the attack	

Since: 3.0.0

ags_recycling_new ()

```
AgsRecycling~*
ags_recycling_new (GObject *output_soundcard);
```

Creates a AgsRecycling, with defaults of *soundcard* .

Parameters

output_soundcard	the GObject implementing AgsSoundcard	

Returns

a new AgsRecycling

Since: 3.0.0

AGS_IS_RECYCLING()

```
#define AGS_IS_RECYCLING(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECYCLING))
```

AGS_IS_RECYCLING_CLASS()

```
#define AGS_IS_RECYCLING_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECYCLING))
```

AGS_RECYCLING()

```
#define AGS_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECYCLING, AgsRecycling))
```

AGS_RECYCLING_CLASS()

```
#define AGS_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECYCLING, AgsRecyclingClass))
```

AGS_RECYCLING_GET_CLASS()

```
#define AGS_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECYCLING, AgsRecyclingClass))
```

ags_recycling_flags_get_type ()

```
GType
ags_recycling_flags_get_type ();
```

ags_recycling_get_type ()

```
GType
ags_recycling_get_type (void);
```

Types and Values

enum AgsRecyclingFlags

Enum values to control the behavior or indicate internal state of AgsRecycling by enable/disable as flags.

Members

AGS_RECYCLING_MUTED	recycling is muted

AGS_TYPE_RECYCLING

```
#define AGS_TYPE_RECYCLING                (ags_recycling_get_type())
```

struct AgsRecycling

```
struct AgsRecycling;
```

struct AgsRecyclingClass

```
struct AgsRecyclingClass {
  GObjectClass gobject;

  void (*add_audio_signal)(AgsRecycling *recycling,
         AgsAudioSignal *audio_signal);
  void (*remove_audio_signal)(AgsRecycling *recycling,
           AgsAudioSignal *audio_signal);

  void (*data_request)(AgsRecycling *recycling,
          AgsAudioSignal *audio_signal);
};
```

Property Details

The "audio-signal" property

```
  "audio-signal"              gpointer
```

The containing AgsAudioSignal.

[transfer full]

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "buffer-size" property

```
  "buffer-size"               guint
```

The buffer size.

Owner: AgsRecycling

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "channel" property

```
  "channel"                   AgsChannel~*
```

The assigned AgsChannel.

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The format.

Owner: AgsRecycling

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "input-soundcard" property

```
"input-soundcard"           GObject~*
```

The assigned input soundcard.

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "input-soundcard-channel" property

```
"input-soundcard-channel"   int
```

The input soundcard channel.

Owner: AgsRecycling

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "next" property

```
"next"                      AgsRecycling~*
```

The assigned next AgsRecycling.

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "output-soundcard" property

```
"output-soundcard"          GObject~*
```

The assigned output soundcard acting as default sink.

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "output-soundcard-channel" property

```
"output-soundcard-channel" int
```

The output soundcard channel.

Owner: AgsRecycling

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "parent" property

```
"parent"                    AgsRecycling~*
```

The assigned parent AgsRecycling.

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "prev" property

```
"prev"                      AgsRecycling~*
```

The assigned prev AgsRecycling.

Owner: AgsRecycling

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate.

Owner: AgsRecycling

Flags: Read / Write

Default value: 0

Since: 3.0.0

Signal Details

The "add-audio-signal" signal

```
void
user_function (AgsRecycling *recycling,
               GObject      *audio_signal,
               gpointer      user_data)
```

The ::add-audio-signal signal is emited as adding AgsAudioSignal.

Parameters

recycling	an AgsRecycling	
audio_signal	the AgsAudioSignal to add	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "data-request" signal

```
void
user_function (AgsRecycling *recycling,
               GObject      *audio_signal,
               gpointer      user_data)
```

The ::data-request signal is emited as requesting data for *audio_signal* .

Parameters

recycling	an AgsRecycling	
audio_signal	the AgsAudioSignal to request	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "remove-audio-signal" signal

```
void
user_function (AgsRecycling *recycling,
               GObject      *audio_signal,
               gpointer      user_data)
```

The ::remove-audio-signal signal is emited as removing AgsAudioSignal.

Parameters

recycling	an AgsRecycling	
audio_signal	the AgsAudioSignal to remove	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

9.8 AgsAudioSignal

AgsAudioSignal — Contains the audio data and its alignment

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_SIGNAL_GET_OBJ_MUTEX()
#define	AGS_AUDIO_SIGNAL_GET_STREAM_MUTEX()
GRecMutex *	ags_audio_signal_get_obj_mutex ()
void	ags_audio_signal_stream_lock ()
void	ags_audio_signal_stream_unlock ()
gboolean	ags_audio_signal_test_flags ()
void	ags_audio_signal_set_flags ()
void	ags_audio_signal_unset_flags ()
void *	ags_stream_alloc ()
void	ags_stream_free ()
void *	ags_stream_slice_alloc ()
void	ags_stream_slice_free ()
GObject *	ags_audio_signal_get_recycling ()
void	ags_audio_signal_set_recycling ()
GObject *	ags_audio_signal_get_output_soundcard ()
void	ags_audio_signal_set_output_soundcard ()
GObject *	ags_audio_signal_get_input_soundcard ()
void	ags_audio_signal_set_input_soundcard ()
guint	ags_audio_signal_get_samplerate ()
void	ags_audio_signal_set_samplerate ()
guint	ags_audio_signal_get_buffer_size ()
void	ags_audio_signal_set_buffer_size ()
guint	ags_audio_signal_get_format ()
void	ags_audio_signal_set_format ()
GList *	ags_audio_signal_get_note ()
void	ags_audio_signal_set_note ()
void	ags_audio_signal_add_note ()
void	ags_audio_signal_remove_note ()
GList *	ags_audio_signal_get_stream ()
void	ags_audio_signal_set_stream ()
void	ags_audio_signal_refresh_data ()
void	ags_audio_signal_add_stream ()
void	ags_audio_signal_stream_resize ()
void	ags_audio_signal_stream_safe_resize ()
void	ags_audio_signal_clear ()
void	ags_audio_signal_duplicate_stream ()
void	ags_audio_signal_feed ()
void	ags_audio_signal_feed_extended ()
void	ags_audio_signal_open_feed ()
void	ags_audio_signal_continue_feed ()
void	ags_audio_signal_close_feed ()
guint	ags_audio_signal_get_length_till_current ()
gboolean	ags_audio_signal_contains_note ()
AgsAudioSignal *	ags_audio_signal_get_template ()

GList *	ags_audio_signal_get_rt_template ()
GList *	ags_audio_signal_find_stream_current ()
GList *	ags_audio_signal_find_by_recall_id ()
gboolean	ags_audio_signal_is_active ()
AgsAudioSignal *	ags_audio_signal_new ()
AgsAudioSignal *	ags_audio_signal_new_with_length ()
#define	AGS_AUDIO_SIGNAL()
#define	AGS_AUDIO_SIGNAL_CLASS()
#define	AGS_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_AUDIO_SIGNAL()
#define	AGS_IS_AUDIO_SIGNAL_CLASS()
GType	ags_audio_signal_flags_get_type ()
GType	ags_audio_signal_get_type ()

Properties

guint	attack	Read / Write
guint	buffer-size	Read / Write
AgsComplex *	damping	Read / Write
double	delay	Read / Write
guint	first-frame	Read / Write
guint	format	Read / Write
guint	frame-count	Read / Write
GObject *	input-soundcard	Read / Write
int	input-soundcard-channel	Read / Write
guint	last-frame	Read / Write
guint	length	Read / Write
guint	loop-end	Read / Write
guint	loop-start	Read / Write
gpointer	note	Read / Write
GObject *	output-soundcard	Read / Write
int	output-soundcard-channel	Read / Write
AgsRecallID *	recall-id	Read / Write
AgsRecycling *	recycling	Read / Write
AgsAudioSignal *	rt-template	Read / Write
guint	samplerate	Read / Write
gpointer	stream	Read
gpointer	stream-current	Read
gpointer	stream-end	Read
AgsAudioSignal *	template	Read / Write
guint	timbre-end	Read / Write
guint	timbre-start	Read / Write
AgsComplex *	vibration	Read / Write
guint	word-size	Read

Signals

void	add-note	Run Last
void	refresh-data	Run Last
void	remove-note	Run Last

Types and Values

enum	AgsAudioSignalFlags
#define	AGS_TYPE_AUDIO_SIGNAL
struct	AgsAudioSignal
struct	AgsAudioSignalClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsAudioSignalFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsAudioSignal
```

Implemented Interfaces

AgsAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/ags_audio_signal.h>
```

Description

AgsAudioSignal organizes audio data within a GList whereby data pointing to the buffer.

It has `output-soundcard` and `input-soundcard` properties AgsRecall implementations should default to.

The `stream` property contains the processing audio data. You may set `loop-start` and `loop-end` to loop over samples until desired audio buffer length is achieved.

As accessing the audio data you should lock it with ags_audio_signal_stream_lock() and as completed your work you release the lock with ags_audio_signal_stream_unlock().

The `note` property contains information about assigned AgsNote objects coming from AgsNotation residing in AgsAudio.

As you might see multiple key presses may apply to one AgsAudioSignal.

Functions

AGS_AUDIO_SIGNAL_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_SIGNAL_GET_OBJ_MUTEX(obj) (&(((AgsAudioSignal *) obj)->obj_mutex))
```

AGS_AUDIO_SIGNAL_GET_STREAM_MUTEX()

```
#define AGS_AUDIO_SIGNAL_GET_STREAM_MUTEX(obj) (&(((AgsAudioSignal *) obj)->stream_mutex))
```

ags_audio_signal_get_obj_mutex ()

```
GRecMutex~*
ags_audio_signal_get_obj_mutex (AgsAudioSignal *audio_signal);
```

Get object mutex.

Parameters

audio_signal	the AgsAudioSignal

Returns

the GRecMutex to lock *audio_signal*

Since: 3.1.0

ags_audio_signal_stream_lock ()

```
void
ags_audio_signal_stream_lock (AgsAudioSignal *audio_signal);
```

Lock stream mutex.

Parameters

audio_signal	the AgsAudioSignal

Since: 3.1.0

ags_audio_signal_stream_unlock ()

```
void
ags_audio_signal_stream_unlock (AgsAudioSignal *audio_signal);
```

Unlock stream mutex.

Parameters

audio_signal	the AgsAudioSignal

Since: 3.1.0

ags_audio_signal_test_flags ()

```
gboolean
ags_audio_signal_test_flags (AgsAudioSignal *audio_signal,
                             guint flags);
```

Test *flags* to be set on *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_signal_set_flags ()

```
void
ags_audio_signal_set_flags (AgsAudioSignal *audio_signal,
                            guint flags);
```

Enable a feature of *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	
flags	see AgsAudioSignalFlags	

Since: 3.0.0

ags_audio_signal_unset_flags ()

```
void
ags_audio_signal_unset_flags (AgsAudioSignal *audio_signal,
                              guint flags);
```

Disable a feature of *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	
flags	see AgsAudioSignalFlags	

Since: 3.0.0

ags_stream_alloc ()

```
void~*
ags_stream_alloc (guint buffer_size,
                  guint format);
```

Allocs an audio buffer.

Parameters

buffer_size	the buffer size	
format	the format	

Returns

the audio data array

Since: 3.0.0

ags_stream_free ()

```
void
ags_stream_free (void *buffer);
```

Frees an audio buffer.

Parameters

buffer	the buffer	

Since: 3.0.0

ags_stream_slice_alloc ()

```
void~*
ags_stream_slice_alloc (guint buffer_size,
                        guint format);
```

Allocs an audio buffer.

Parameters

buffer_size	the buffer size	
format	the format	

Returns

the audio data array

Since: 3.3.0

ags_stream_slice_free ()

```
void
ags_stream_slice_free (guint buffer_size,
                       guint format,
                       void *buffer);
```

Frees an audio buffer.

Parameters

buffer_size	the buffer size	
format	the format	
buffer	the buffer	

Since: 3.3.0

ags_audio_signal_get_recycling ()

```
GObject~*
ags_audio_signal_get_recycling (AgsAudioSignal *audio_signal);
```

Get recycling.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the AgsRecycling.

[transfer full]

Since: 3.1.0

ags_audio_signal_set_recycling ()

```
void
ags_audio_signal_set_recycling (AgsAudioSignal *audio_signal,
                                GObject *recycling);
```

Set recycling.

Parameters

audio_signal	the AgsAudioSignal	
recycling	the AgsRecycling	

Since: 3.1.0

ags_audio_signal_get_output_soundcard ()

```
GObject~*
ags_audio_signal_get_output_soundcard (AgsAudioSignal *audio_signal);
```

Get the output soundcard object of *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	

Returns

the output soundcard.

[transfer full]

Since: 3.1.0

ags_audio_signal_set_output_soundcard ()

```
void
ags_audio_signal_set_output_soundcard (AgsAudioSignal *audio_signal,
                                       GObject *output_soundcard);
```

Set the output soundcard object of *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	
output_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_audio_signal_get_input_soundcard ()

```
GObject~*
ags_audio_signal_get_input_soundcard (AgsAudioSignal *audio_signal);
```

Get the input soundcard object of *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	

Returns

the input soundcard.

[transfer full]

Since: 3.1.0

ags_audio_signal_set_input_soundcard ()

```
void
ags_audio_signal_set_input_soundcard (AgsAudioSignal *audio_signal,
                                      GObject *input_soundcard);
```

Set the input soundcard object of *audio_signal* .

Parameters

audio_signal	an AgsAudioSignal	
input_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_audio_signal_get_samplerate ()

```
guint
ags_audio_signal_get_samplerate (AgsAudioSignal *audio_signal);
```

Gets samplerate.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the samplerate

Since: 3.1.0

ags_audio_signal_set_samplerate ()

```
void
ags_audio_signal_set_samplerate (AgsAudioSignal *audio_signal,
                                 guint samplerate);
```

Set samplerate.

Parameters

audio_signal	the AgsAudioSignal	
samplerate	the samplerate	

Since: 3.0.0

ags_audio_signal_get_buffer_size ()

```
guint
ags_audio_signal_get_buffer_size (AgsAudioSignal *audio_signal);
```

Gets buffer size.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the buffer size

Since: 3.1.0

ags_audio_signal_set_buffer_size ()

```
void
ags_audio_signal_set_buffer_size (AgsAudioSignal *audio_signal,
                                  guint buffer_size);
```

Set buffer size.

Parameters

audio_signal	the AgsAudioSignal	
buffer_size	the buffer size	

Since: 3.0.0

ags_audio_signal_get_format ()

```
guint
ags_audio_signal_get_format (AgsAudioSignal *audio_signal);
```

Gets format.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the format

Since: 3.1.0

ags_audio_signal_set_format ()

```
void
ags_audio_signal_set_format (AgsAudioSignal *audio_signal,
                             guint format);
```

Set format.

Parameters

audio_signal	the AgsAudioSignal	
format	the format	

Since: 3.0.0

ags_audio_signal_get_note ()

```
GList~*
ags_audio_signal_get_note (AgsAudioSignal *audio_signal);
```

Get note.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the GList containig AgsNote.

[element-type AgsAudio.Note][transfer full]

Since: 3.1.0

ags_audio_signal_set_note ()

```
void
ags_audio_signal_set_note (AgsAudioSignal *audio_signal,
                           GList *note);
```

Set note by replacing existing.

Parameters

audio_signal	the AgsAudioSignal	
note	the GList containing AgsNote.	*[element-type AgsAudio.Note][transfer full]*

Since: 3.1.0

ags_audio_signal_add_note ()

```
void
ags_audio_signal_add_note (AgsAudioSignal *audio_signal,
                           GObject *note);
```

Add note to `audio_signal` .

Parameters

audio_signal	the AgsAudioSignal	
note	the AgsNote	

Since: 3.0.0

ags_audio_signal_remove_note ()

```
void
ags_audio_signal_remove_note (AgsAudioSignal *audio_signal,
                              GObject *note);
```

Remove note from *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	
note	the AgsNote	

Since: 3.0.0

ags_audio_signal_get_stream ()

```
GList~*
ags_audio_signal_get_stream (AgsAudioSignal *audio_signal);
```

Get stream.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the GList containig audio data.

[element-type guint8][transfer none]

Since: 3.8.11

ags_audio_signal_set_stream ()

```
void
ags_audio_signal_set_stream (AgsAudioSignal *audio_signal,
                             GList *stream);
```

Set stream by replacing existing.

Parameters

audio_signal	the AgsAudioSignal	
stream	the GList containing audio data.	*[element-type guint8][transfer full]*

Since: 3.8.11

ags_audio_signal_refresh_data ()

```
void
ags_audio_signal_refresh_data (AgsAudioSignal *audio_signal);
```

ags_audio_signal_add_stream ()

```
void
ags_audio_signal_add_stream (AgsAudioSignal *audio_signal);
```

Adds a buffer at the end of the stream.

Parameters

audio_signal	the AgsAudioSignal	

Since: 3.0.0

ags_audio_signal_stream_resize ()

```
void
ags_audio_signal_stream_resize (AgsAudioSignal *audio_signal,
                                guint length);
```

Resize stream of *audio_signal* to *length* number of buffers.

Parameters

audio_signal	the AgsAudioSignal	
length	the new length	

Since: 3.0.0

ags_audio_signal_stream_safe_resize ()

```
void
ags_audio_signal_stream_safe_resize (AgsAudioSignal *audio_signal,
                                     guint length);
```

Resize stream of *audio_signal* to *length* number of buffers. But doesn't shrink more than the current stream position.

Parameters

audio_signal	the AgsAudioSignal	
length	the new length	

Since: 3.0.0

ags_audio_signal_clear ()

```
void
ags_audio_signal_clear (AgsAudioSignal *audio_signal);
```

Clear *audio_signal* .

Parameters

audio_signal	the AgsAudioSignal	

Since: 3.4.1

ags_audio_signal_duplicate_stream ()

```
void
ags_audio_signal_duplicate_stream (AgsAudioSignal *audio_signal,
                                   AgsAudioSignal *template);
```

Apply *template* audio data to *audio_signal* . Note should only be invoked by proper recall context because only the stream of *template* is locked.

Parameters

audio_signal	the AgsAudioSignal	
template	the template AgsAudioSignal	

Since: 3.0.0

ags_audio_signal_feed ()

```
void
ags_audio_signal_feed (AgsAudioSignal *audio_signal,
                       AgsAudioSignal *template,
                       guint frame_count);
```

Feed audio signal to grow upto frame count.

Parameters

audio_signal	the AgsAudioSignal	
template	the template AgsAudioSignal	
frame_count	the new frame count	

Since: 3.0.0

ags_audio_signal_feed_extended ()

```
void
ags_audio_signal_feed_extended (AgsAudioSignal *audio_signal,
                                AgsAudioSignal *template,
                                guint frame_count,
                                guint old_frame_count,
                                gboolean do_open,
                                gboolean do_close);
```

Feed audio signal to grow upto frame count.

Parameters

audio_signal	the AgsAudioSignal	
template	the template AgsAudioSignal	
frame_count	the new frame count	
old_frame_count	the old frame count	
do_open	open feed	
do_close	close feed	

Since: 3.3.0

ags_audio_signal_open_feed ()

```
void
ags_audio_signal_open_feed (AgsAudioSignal *audio_signal,
                            AgsAudioSignal *template,
                            guint frame_count,
                            guint old_frame_count);
```

Feed audio signal to grow upto frame count.

Parameters

audio_signal	the AgsAudioSignal	
template	the template AgsAudioSignal	
frame_count	the new frame count	
old_frame_count	the old frame count	

Since: 3.3.0

ags_audio_signal_continue_feed ()

```
void
ags_audio_signal_continue_feed (AgsAudioSignal *audio_signal,
                                AgsAudioSignal *template,
                                guint frame_count,
                                guint old_frame_count);
```

Feed audio signal to grow upto frame count.

Parameters

audio_signal	the AgsAudioSignal	
template	the template AgsAudioSignal	
frame_count	the new frame count	
old_frame_count	the old frame count	

Since: 3.3.0

ags_audio_signal_close_feed ()

```
void
ags_audio_signal_close_feed (AgsAudioSignal *audio_signal,
                             AgsAudioSignal *template,
                             guint frame_count,
                             guint old_frame_count);
```

Feed audio signal to grow upto frame count.

Parameters

audio_signal	the AgsAudioSignal	
template	the template AgsAudioSignal	
frame_count	the new frame count	
old_frame_count	the old frame count	

Since: 3.3.0

ags_audio_signal_get_length_till_current ()

```
guint
ags_audio_signal_get_length_till_current
                                (AgsAudioSignal *audio_signal);
```

Counts the buffers from :stream upto :stream-current.

Parameters

audio_signal	the AgsAudioSignal	

Returns

the counted length.

Since: 3.0.0

ags_audio_signal_contains_note ()

```
gboolean
ags_audio_signal_contains_note (AgsAudioSignal *audio_signal,
                                AgsNote *note);
```

Check if *audio_signal* contains *note* .

Parameters

audio_signal	the AgsAudioSignal	
note	the AgsNote	

Returns

TRUE on success, otherwise FALSE

Since: 3.3.0

ags_audio_signal_get_template ()

```
AgsAudioSignal~*
ags_audio_signal_get_template (GList *audio_signal);
```

Retrieve the template audio signal.

Parameters

audio_signal	the GList containing AgsAudioSignal.	*[element-type AgsAudio.AudioSignal][transfer none]*

Returns

the template AgsAudioSignal or NULL if not found.

[transfer full]

Since: 3.0.0

ags_audio_signal_get_rt_template ()

```
GList~*
ags_audio_signal_get_rt_template (GList *audio_signal);
```

Retrieve the realtime template audio signal.

Parameters

audio_signal	a GList containing AgsAudioSignal.	*[element-type AgsAudio.AudioSignal][transfer none]*

Returns

the rt-templates as GList containing AgsAudioSignal.

[element-type AgsAudio.AudioSignal][transfer full]

Since: 3.0.0

ags_audio_signal_find_stream_current ()

```
GList~*
ags_audio_signal_find_stream_current (GList *audio_signal,
                                      GObject *recall_id);
```

Retrieve next current stream of AgsAudioSignal list. Warning this function does not lock the stream mutex.

Parameters

audio_signal	the GList containing AgsAudioSignal.	*[element-type AgsAudio.AudioSignal][transfer none]*
recall_id	the matching AgsRecallID	

Returns

next GList matching AgsRecallID.

[element-type AgsAudio.AudioSignal][transfer none]

Since: 3.0.0

ags_audio_signal_find_by_recall_id ()

```
GList~*
ags_audio_signal_find_by_recall_id (GList *audio_signal,
                                    GObject *recall_id);
```

Retrieve next audio signal refering to `recall_id`

Parameters

audio_signal	a GList containing AgsAudioSignal.	*[element-type AgsAudio.AudioSignal][transfer none]*
recall_id	matching AgsRecallID	

Returns

matching GList containing AgsAudioSignal.

[element-type AgsAudio.AudioSignal][transfer none]

Since: 3.0.0

ags_audio_signal_is_active ()

```
gboolean
ags_audio_signal_is_active (GList *audio_signal,
                            GObject *recall_id);
```

Check if is active.

Parameters

audio_signal	the GList containing AgsAudioSignal.	*[element-type AgsAudio.AudioSignal][transfer none]*
recall_id	the AgsRecallID	

Returns

TRUE if related audio signal to recall id is available, otherwise FALSE

Since: 3.0.0

ags_audio_signal_new ()

```
AgsAudioSignal~*
ags_audio_signal_new (GObject *output_soundcard,
                      GObject *recycling,
                      GObject *recall_id);
```

Creates a AgsAudioSignal, with defaults of `output_soundcard` , linking `recycling` tree and refering to `recall_id` .

Parameters

output_soundcard	the assigned output AgsSoundcard	
recycling	the AgsRecycling	
recall_id	the AgsRecallID, it can be NULL if AGS_AUDIO_SIGNAL_TEMPLATE is set	

Returns

a new AgsAudioSignal

Since: 3.0.0

ags_audio_signal_new_with_length ()

```
AgsAudioSignal~*
ags_audio_signal_new_with_length (GObject *output_soundcard,
                                  GObject *recycling,
                                  GObject *recall_id,
                                  guint length);
```

Creates a AgsAudioSignal, with defaults of `soundcard` , linking `recycling` tree and refering to `recall_id` . The audio data is tiled to `length` frame count.

Parameters

output_soundcard	the assigned output AgsSoundcard	
recycling	the AgsRecycling	
recall_id	the AgsRecallID, it can be NULL if AGS_AUDIO_SIGNAL_TEMPLATE is set	
length	audio data frame count	

Returns

a new AgsAudioSignal

Since: 3.0.0

AGS_AUDIO_SIGNAL()

```
#define AGS_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_SIGNAL, AgsAudioSignal))
```

AGS_AUDIO_SIGNAL_CLASS()

```
#define AGS_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_AUDIO_SIGNAL, AgsAudioSignalClass))
```

AGS_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_AUDIO_SIGNAL, AgsAudioSignalClass))
```

AGS_IS_AUDIO_SIGNAL()

```
#define AGS_IS_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_SIGNAL))
```

AGS_IS_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_SIGNAL))
```

ags_audio_signal_flags_get_type ()

```
GType
ags_audio_signal_flags_get_type ();
```

ags_audio_signal_get_type ()

```
GType
ags_audio_signal_get_type ();
```

Types and Values

enum AgsAudioSignalFlags

Enum values to control the behavior or indicate internal state of AgsAudioSignal by enable/disable as flags.

Members

AGS_AUDIO_SIGNAL_TEMPLATE	the audio signal acts as a template
AGS_AUDIO_SIGNAL_RT_TEMPLATE	the audio signal acts as a real-time template
AGS_AUDIO_SIGNAL_MASTER	the audio signal needs master
AGS_AUDIO_SIGNAL_FEED	the audio signal needs feed

AGS_AUDIO_SIGNAL_RECYCLED	the audio signal is recycled
AGS_AUDIO_SIGNAL_STREAM	the audio signal needs stream
AGS_AUDIO_SIGNAL_SLICE_ALLOC	the audio signal allocates from slice

AGS_TYPE_AUDIO_SIGNAL

```
#define AGS_TYPE_AUDIO_SIGNAL                (ags_audio_signal_get_type())
```

struct AgsAudioSignal

```
struct AgsAudioSignal;
```

struct AgsAudioSignalClass

```
struct AgsAudioSignalClass {
  GObjectClass gobject;

  void (*add_note)(AgsAudioSignal *audio_signal,
       GObject *note);
  void (*remove_note)(AgsAudioSignal *audio_signal,
          GObject *note);

  void (*refresh_data)(AgsAudioSignal *audio_signal);
};
```

Property Details

The "attack" property

```
"attack"                    guint
```

The attack to be used.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size to be used.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "damping" property

```
"damping"                   AgsComplex~*
```

Damping of timbre.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "delay" property

```
"delay"                     double
```

The delay to be used.

Owner: AgsAudioSignal

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "first-frame" property

```
"first-frame"               guint
```

The first frame of stream.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The format to be used.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "frame-count" property

```
"frame-count"               guint
```

The initial size of audio data.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "input-soundcard" property

```
"input-soundcard"           GObject~*
```

The assigned input AgsSoundcard.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "input-soundcard-channel" property

```
"input-soundcard-channel"   int
```

The input soundcard channel.

Owner: AgsAudioSignal

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "last-frame" property

```
"last-frame"                guint
```

The last frame of stream.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "length" property

```
"length"                    guint
```

The length of the stream.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "loop-end" property

```
"loop-end"                  guint
```

The loop end of stream.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "loop-start" property

```
"loop-start"                guint
```

The loop start of stream.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "note" property

```
"note"                      gpointer
```

The assigned AgsNote providing default settings.

[transfer full]

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "output-soundcard" property

```
"output-soundcard"          GObject~*
```

The assigned output AgsSoundcard providing default settings.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "output-soundcard-channel" property

```
"output-soundcard-channel" int
```

The output soundcard channel.

Owner: AgsAudioSignal

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "recall-id" property

```
"recall-id"                 AgsRecallID~*
```

The assigned AgsRecallID providing context.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "recycling" property

```
"recycling"                 AgsRecycling~*
```

The assigned AgsRecycling linking tree.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "rt-template" property

```
"rt-template"               AgsAudioSignal~*
```

The assigned AgsAudioSignal realtime template.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "stream" property

```
"stream"                    gpointer
```

The stream it contains.

Owner: AgsAudioSignal

Flags: Read

Since: 3.0.0

The "stream-current" property

```
"stream-current"            gpointer
```

The current stream.

Owner: AgsAudioSignal

Flags: Read

Since: 3.0.0

The "stream-end" property

```
"stream-end"                gpointer
```

The end of stream.

Owner: AgsAudioSignal

Flags: Read

Since: 3.0.0

The "template" property

```
"template"                  AgsAudioSignal~*
```

The assigned AgsAudioSignal template.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "timbre-end" property

```
"timbre-end"                guint
```

The timbre's end frame.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "timbre-start" property

```
"timbre-start"              guint
```

The timbre's start frame.

Owner: AgsAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "vibration" property

```
"vibration"                 AgsComplex~*
```

Vibration of timbre.

Owner: AgsAudioSignal

Flags: Read / Write

Since: 3.0.0

The "word-size" property

```
"word-size"                 guint
```

The word size of frame.

Owner: AgsAudioSignal

Flags: Read

Default value: 0

Since: 3.0.0

Signal Details

The "add-note" signal

```
void
user_function (AgsAudioSignal *audio_signal,
               GObject        *note,
               gpointer        user_data)
```

The ::add-note signal notifies about adding *note* .

Parameters

audio_signal	the AgsAudioSignal	
note	the AgsNote	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "refresh-data" signal

```
void
user_function (AgsAudioSignal *audio_signal,
               gpointer        user_data)
```

The ::refresh-data signal notifies about requesting to refresh data.

Parameters

audio_signal	the AgsAudioSignal	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "remove-note" signal

```
void
user_function (AgsAudioSignal *audio_signal,
               GObject        *note,
               gpointer        user_data)
```

The ::remove-note signal notifies about removing *note* .

Parameters

audio_signal	the AgsAudioSignal	
note	the AgsNote	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

9.9 AgsPort

AgsPort — Perform thread-safe operations

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PORT_GET_OBJ_MUTEX()
gboolean	ags_port_test_flags ()
void	ags_port_set_flags ()
void	ags_port_unset_flags ()
void	ags_port_safe_read ()
void	ags_port_safe_read_raw ()
void	ags_port_safe_write ()
void	ags_port_safe_write_raw ()
void	ags_port_safe_get_property ()
void	ags_port_safe_set_property ()
GList *	ags_port_find_specifier ()
GList *	ags_port_find_plugin_port ()
void	ags_port_add_automation ()
void	ags_port_remove_automation ()
AgsPort *	ags_port_new ()
#define	AGS_IS_PORT()
#define	AGS_IS_PORT_CLASS()
#define	AGS_PORT()
#define	AGS_PORT_CLASS()
#define	AGS_PORT_GET_CLASS()
GType	ags_port_flags_get_type ()
GType	ags_port_get_type ()

Properties

gpointer	automation	Read / Write
char *	control-port	Read / Write
AgsConversion *	conversion	Read / Write
char *	plugin-name	Read / Write
AgsPluginPort *	plugin-port	Read / Write
gboolean	port-value-is-pointer	Read / Write
guint	port-value-length	Read / Write
guint	port-value-size	Read / Write
GType *	port-value-type	Read / Write
char *	specifier	Read / Write

Signals

void	safe-get-property	Run Last
void	safe-read	Run Last
void	safe-set-property	Run Last
void	safe-write	Run Last

Types and Values

enum	AgsPortFlags
#define	AGS_TYPE_PORT
struct	AgsPort
struct	AgsPortClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsPortFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsPort
```

Implemented Interfaces

AgsPort implements AgsConnectable.

Includes

```
#include <ags/audio/ags_port.h>
```

Description

AgsPort provides a thread-safe way to access or change values or properties.

The `specifier` property has to be unique within context.

You call ags_port_safe_write() to alter a value or ags_port_safe_read() to read a port.

Functions

AGS_PORT_GET_OBJ_MUTEX()

```
#define AGS_PORT_GET_OBJ_MUTEX(obj) (&(((AgsPort *) obj)->obj_mutex))
```

ags_port_test_flags ()

```
gboolean
ags_port_test_flags (AgsPort *port,
                     guint flags);
```

Test *flags* to be set on *port* .

Parameters

port	the AgsPort	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_port_set_flags ()

```
void
ags_port_set_flags (AgsPort *port,
                    guint flags);
```

Set flags.

Parameters

port	the AgsPort	
flags	the flags	

Since: 3.0.0

ags_port_unset_flags ()

```
void
ags_port_unset_flags (AgsPort *port,
                      guint flags);
```

Unset flags.

Parameters

port	the AgsPort	
flags	the flags	

Since: 3.0.0

ags_port_safe_read ()

```
void
ags_port_safe_read (AgsPort *port,
                    GValue *value);
```

Perform safe read.

Parameters

port	an AgsPort	
value	the GValue to store result	

Since: 3.0.0

ags_port_safe_read_raw ()

```
void
ags_port_safe_read_raw (AgsPort *port,
                        GValue *value);
```

Perform safe read.

Parameters

port	an AgsPort	
value	the GValue to store result	

Since: 3.0.0

ags_port_safe_write ()

```
void
ags_port_safe_write (AgsPort *port,
                     GValue *value);
```

Perform safe write.

Parameters

port	an AgsPort	
value	the GValue containing data	

Since: 3.0.0

ags_port_safe_write_raw ()

```
void
ags_port_safe_write_raw (AgsPort *port,
                         GValue *value);
```

ags_port_safe_get_property ()

```
void
ags_port_safe_get_property (AgsPort *port,
                            gchar *property_name,
                            GValue *value);
```

Perform safe get property.

Parameters

port	an AgsPort	
property_name	the property's name	
value	the GValue to store the result	

Since: 3.0.0

ags_port_safe_set_property ()

```
void
ags_port_safe_set_property (AgsPort *port,
                            gchar *property_name,
                            GValue *value);
```

Perform safe set property.

Parameters

port	an AgsPort	
property_name	the property's name	
value	the GValue containing data	

Since: 3.0.0

ags_port_find_specifier ()

```
GList~*
ags_port_find_specifier (GList *port,
                         gchar *specifier);
```

Retrieve port by specifier.

Parameters

port	the GList containing AgsPort.	*[element-type AgsAudio.Port][transfer none]*
specifier	the recall specifier to match	

Returns

Next matching GList or NULL.

[element-type AgsAudio.Port][transfer none]

Since: 3.0.0

ags_port_find_plugin_port ()

```
GList~*
ags_port_find_plugin_port (GList *port,
                           GObject *plugin_port);
```

Retrieve port by plugin port.

Parameters

port	the GList containing AgsPort.	*[element-type AgsAudio.Port][transfer none]*
plugin_port	the AgsPluginPort	

Returns

Next matching GList or NULL.

[element-type AgsAudio.Port][transfer none]

Since: 3.11.4

ags_port_add_automation ()

```
void
ags_port_add_automation (AgsPort *port,
                         GObject *automation);
```

Adds an automation.

Parameters

port	the AgsPort	
automation	the AgsAutomation	

Since: 3.0.0

ags_port_remove_automation ()

```
void
ags_port_remove_automation (AgsPort *port,
                            GObject *automation);
```

Removes an automation.

Parameters

port	the AgsPort	
automation	the AgsAutomation	

Since: 3.0.0

ags_port_new ()

```
AgsPort~*
ags_port_new ();
```

Creates an AgsPort.

Returns

a new AgsPort.

Since: 3.0.0

AGS_IS_PORT()

```
#define AGS_IS_PORT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_PORT))
```

AGS_IS_PORT_CLASS()

```
#define AGS_IS_PORT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_PORT))
```

AGS_PORT()

```
#define AGS_PORT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_PORT,  ↩
    AgsPort))
```

AGS_PORT_CLASS()

```
#define AGS_PORT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_PORT,  ↩
    AgsPortClass))
```

AGS_PORT_GET_CLASS()

```
#define AGS_PORT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_PORT,  ↩
    AgsPortClass))
```

ags_port_flags_get_type ()

```
GType
ags_port_flags_get_type ();
```

ags_port_get_type ()

```
GType
ags_port_get_type ();
```

Types and Values

enum AgsPortFlags

Enum values to control the behavior or indicate internal state of AgsPort by enable/disable as flags.

Members

AGS_PORT_CONVERT_ALWAYS	convert always
AGS_PORT_USE_LADSPA_FLOAT	use ladspa float
AGS_PORT_IS_OUTPUT	is output
AGS_PORT_INFINITE_RANGE	infinite range

AGS_TYPE_PORT

```
#define AGS_TYPE_PORT                (ags_port_get_type())
```

struct AgsPort

```
struct AgsPort;
```

struct AgsPortClass

```
struct AgsPortClass {
  GObjectClass gobject;

  void (*safe_read)(AgsPort *port, GValue *value);
  void (*safe_write)(AgsPort *port, GValue *value);

  void (*safe_get_property)(AgsPort *port, gchar *property_name, GValue *value);
  void (*safe_set_property)(AgsPort *port, gchar *property_name, GValue *value);
};
```

Property Details

The "automation" property

```
  "automation"                gpointer
```

The port's automation.

[transfer full]

Owner: AgsPort

Flags: Read / Write

Since: 3.0.0

The "control-port" property

```
  "control-port"              char~*
```

The assigned plugin control port.

Owner: AgsPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "conversion" property

```
  "conversion"                AgsConversion~*
```

The port's conversion object.

Owner: AgsPort

Flags: Read / Write

Since: 3.0.0

The "plugin-name" property

```
"plugin-name"                char~*
```

The assigned plugin.

Owner: AgsPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "plugin-port" property

```
"plugin-port"                AgsPluginPort~*
```

The plugin-port.

Owner: AgsPort

Flags: Read / Write

Since: 3.0.0

The "port-value-is-pointer" property

```
"port-value-is-pointer"      gboolean
```

Specify port data as pointer.

Owner: AgsPort

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "port-value-length" property

```
"port-value-length"          guint
```

The port's data array length.

Owner: AgsPort

Flags: Read / Write

Default value: 1

Since: 3.0.0

The "port-value-size" property

```
"port-value-size"            guint
```

The port's data type size.

Owner: AgsPort

Flags: Read / Write

Allowed values: [1,16]

Default value: 8

Since: 3.0.0

The "port-value-type" property

```
  "port-value-type"          GType~*
```

The port's data type.

Owner: AgsPort

Flags: Read / Write

Allowed values: void

Since: 3.0.0

The "specifier" property

```
  "specifier"                char~*
```

The assigned plugin identifier.

Owner: AgsPort

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Signal Details

The "safe-get-property" signal

```
void
user_function (AgsPort *port,
               char    *property_name,
               gpointer value,
               gpointer user_data)
```

The ::safe-get-property signal is emited while safe get property.

Parameters

port	the object providing safe get property	
property_name	the property name	
value	the GValue	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "safe-read" signal

```
void
user_function (AgsPort *port,
               gpointer value,
               gpointer user_data)
```

The ::safe-read signal is emited while doing safe read operation.

Parameters

port	the object providing safe read	
value	the GValue	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "safe-set-property" signal

```
void
user_function (AgsPort *port,
               char    *property_name,
               gpointer value,
               gpointer user_data)
```

The ::safe-set-property signal is emited while safe set property.

Parameters

port	the object providing safe set property	
property_name	the property name	
value	the GValue	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "safe-write" signal

```
void
user_function (AgsPort *port,
               gpointer value,
               gpointer user_data)
```

The ::safe-write signal is emited while doing safe write operation.

Parameters

port	the object providing safe write	
value	the GValue	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 10

Storage objects

The storage objects cover all editing related operations.

10.1 AgsNotation

AgsNotation — Notation class supporting selection and clipboard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_NOTATION_GET_OBJ_MUTEX()
GRecMutex *	ags_notation_get_obj_mutex ()
gboolean	ags_notation_test_flags ()
void	ags_notation_set_flags ()
void	ags_notation_unset_flags ()
GList *	ags_notation_find_near_timestamp ()
GList *	ags_notation_add ()
gint	ags_notation_sort_func ()
GObject *	ags_notation_get_audio ()
void	ags_notation_set_audio ()
guint	ags_notation_get_audio_channel ()
void	ags_notation_set_audio_channel ()
gboolean	ags_notation_get_is_minor ()
void	ags_notation_set_is_minor ()
guint	ags_notation_get_sharp_flats ()
void	ags_notation_set_sharp_flats ()
AgsTimestamp *	ags_notation_get_timestamp ()
void	ags_notation_set_timestamp ()
GList *	ags_notation_get_note ()
void	ags_notation_set_note ()
void	ags_notation_add_note ()
void	ags_notation_remove_note ()
gboolean	ags_notation_remove_note_at_position ()
GList *	ags_notation_get_selection ()

gboolean	ags_notation_is_note_selected ()
AgsNote *	ags_notation_find_point ()
GList *	ags_notation_find_region ()
GList *	ags_notation_find_offset ()
void	ags_notation_free_selection ()
void	ags_notation_free_all_selection ()
void	ags_notation_add_point_to_selection ()
void	ags_notation_remove_point_from_selection ()
void	ags_notation_add_region_to_selection ()
void	ags_notation_remove_region_from_selection ()
void	ags_notation_add_all_to_selection ()
xmlNode *	ags_notation_copy_selection ()
xmlNode *	ags_notation_cut_selection ()
void	ags_notation_insert_from_clipboard ()
void	ags_notation_insert_from_clipboard_extended ()
guchar *	ags_notation_to_raw_midi ()
AgsNotation *	ags_notation_from_raw_midi ()
AgsNotation *	ags_notation_new ()
#define	AGS_IS_NOTATION()
#define	AGS_IS_NOTATION_CLASS()
#define	AGS_NOTATION()
#define	AGS_NOTATION_CLASS()
#define	AGS_NOTATION_GET_CLASS()
GType	ags_notation_flags_get_type ()
GType	ags_notation_get_type ()

Properties

AgsAudio *	audio	Read / Write
guint	audio-channel	Read / Write
gpointer	note	Read / Write
AgsTimestamp *	timestamp	Read / Write

Types and Values

#define	AGS_NOTATION_DEFAULT_BPM
#define	AGS_NOTATION_TICS_PER_BEAT
#define	AGS_NOTATION_MINIMUM_NOTE_LENGTH
#define	AGS_NOTATION_MAXIMUM_NOTE_LENGTH
#define	AGS_NOTATION_DEFAULT_LENGTH
#define	AGS_NOTATION_DEFAULT_JIFFIE
#define	AGS_NOTATION_DEFAULT_DURATION
#define	AGS_NOTATION_DEFAULT_OFFSET
#define	AGS_NOTATION_DEFAULT_END
#define	AGS_NOTATION_DEFAULT_DIVISION
#define	AGS_NOTATION_CLIPBOARD_VERSION
#define	AGS_NOTATION_CLIPBOARD_TYPE
#define	AGS_NOTATION_CLIPBOARD_FORMAT
enum	AgsNotationFlags
#define	AGS_TYPE_NOTATION
struct	AgsNotation
struct	AgsNotationClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsNotationFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsNotation
```

Includes

```
#include <ags/audio/ags_notation.h>
```

Description

AgsNotation acts as a container of AgsNote. The `timestamp` property tells the engine what the first x offset of AgsNote applies.

You can lookup AgsNotation by start x offset with ags_notation_find_near_timestamp(). The next x offset is calculated as following:

next_x_offset = x_offset + AGS_NOTATION_DEFAULT_OFFSET;

Use ags_notation_add_note() to add AgsNote to AgsNotation and ags_notation_remove_note() to remove it again.

In order to copy or cut notes you select them first by calling ags_notation_add_region_to_selection().

Functions

AGS_NOTATION_GET_OBJ_MUTEX()

```
#define AGS_NOTATION_GET_OBJ_MUTEX(obj) (&(((AgsNotation *) obj)->obj_mutex))
```

ags_notation_get_obj_mutex ()

```
GRecMutex~*
ags_notation_get_obj_mutex (AgsNotation *notation);
```

Get object mutex.

Parameters

notation	the AgsNotation

Returns

the GRecMutex to lock *notation*

Since: 3.1.0

ags_notation_test_flags ()

```
gboolean
ags_notation_test_flags (AgsNotation *notation,
                         guint flags);
```

Test *flags* to be set on *notation* .

Parameters

notation	the AgsNotation	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_notation_set_flags ()

```
void
ags_notation_set_flags (AgsNotation *notation,
                        guint flags);
```

Set *flags* on *notation* .

Parameters

notation	the AgsNotation	
flags	the flags	

Since: 3.0.0

ags_notation_unset_flags ()

```
void
ags_notation_unset_flags (AgsNotation *notation,
                          guint flags);
```

Unset *flags* on *notation* .

Parameters

notation	the AgsNotation	
flags	the flags	

Since: 3.0.0

ags_notation_find_near_timestamp ()

```
GList~*
ags_notation_find_near_timestamp (GList *notation,
                                  guint audio_channel,
                                  AgsTimestamp *timestamp);
```

Retrieve appropriate notation for timestamp.

Parameters

notation	the GList containing AgsNotation.	*[element-type AgsAudio.Notation][transfer none]*
audio_channel	the matching audio channel	
timestamp	the matching timestamp, or NULL to match any timestamp.	*[allow-none]*

Returns

Next matching GList or NULL if not found.

[element-type AgsAudio.Notation][transfer none]

Since: 3.0.0

ags_notation_add ()

```
GList~*
ags_notation_add (GList *notation,
                  AgsNotation *new_notation);
```

Add `new_notation` sorted to `notation`

Parameters

notation	the GList containing AgsNotation.	*[element-type AgsAudio.Notation][transfer none]*
new_notation	the AgsNotation to add	

Returns

the new beginning of `notation` .

[element-type AgsAudio.Notation][transfer none]

Since: 3.0.0

ags_notation_sort_func ()

```
gint
ags_notation_sort_func (gconstpointer a,
                        gconstpointer b);
```

Compare `a` and `b` .

Parameters

a	the AgsNotation	
b	another AgsNotation	

Returns

0 if equal, -1 if smaller and 1 if bigger timestamp

Since: 3.0.0

ags_notation_get_audio ()

```
GObject~*
ags_notation_get_audio (AgsNotation *notation);
```

Get audio.

Parameters

notation	the AgsNotation

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_notation_set_audio ()

```
void
ags_notation_set_audio (AgsNotation *notation,
                        GObject *audio);
```

Set audio.

Parameters

notation	the AgsNotation
audio	the AgsAudio

Since: 3.1.0

ags_notation_get_audio_channel ()

```
guint
ags_notation_get_audio_channel (AgsNotation *notation);
```

Gets audio channel.

Parameters

notation	the AgsNotation

Returns

the audio channel

Since: 3.1.0

ags_notation_set_audio_channel ()

```
void
ags_notation_set_audio_channel (AgsNotation *notation,
                                guint audio_channel);
```

Sets audio channel.

Parameters

notation	the AgsNotation	
audio_channel	the audio channel	

Since: 3.1.0

ags_notation_get_is_minor ()

```
gboolean
ags_notation_get_is_minor (AgsNotation *notation);
```

Gets is minor.

Parameters

notation	the AgsNotation	

Returns

is minor

Since: 3.1.0

ags_notation_set_is_minor ()

```
void
ags_notation_set_is_minor (AgsNotation *notation,
                           gboolean is_minor);
```

Sets is minor.

Parameters

notation	the AgsNotation	
is_minor	is minor	

Since: 3.1.0

ags_notation_get_sharp_flats ()

```
guint
ags_notation_get_sharp_flats (AgsNotation *notation);
```

Gets sharp flats.

Parameters

notation	the AgsNotation

Returns

the sharp flats

Since: 3.1.0

ags_notation_set_sharp_flats ()

```
void
ags_notation_set_sharp_flats (AgsNotation *notation,
                              guint sharp_flats);
```

Sets sharp flats.

Parameters

notation	the AgsNotation
sharp_flats	the sharp flats

Since: 3.1.0

ags_notation_get_timestamp ()

```
AgsTimestamp~*
ags_notation_get_timestamp (AgsNotation *notation);
```

Get timestamp.

Parameters

notation	the AgsNotation

Returns

the AgsTimestamp.

[transfer full]

Since: 3.1.0

ags_notation_set_timestamp ()

```
void
ags_notation_set_timestamp (AgsNotation *notation,
                            AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

notation	the AgsNotation	
timestamp	the AgsTimestamp	

Since: 3.1.0

ags_notation_get_note ()

```
GList~*
ags_notation_get_note (AgsNotation *notation);
```

Get note.

Parameters

notation	the AgsNotation	

Returns

the GList containig AgsNote.

[element-type AgsAudio.Note][transfer full]

Since: 3.1.0

ags_notation_set_note ()

```
void
ags_notation_set_note (AgsNotation *notation,
                       GList *note);
```

Set note by replacing existing.

Parameters

notation	the AgsNotation	
note	the GList containing AgsNote.	*[element-type AgsAudio.Note][transfer full]*

Since: 3.1.0

ags_notation_add_note ()

```
void
ags_notation_add_note (AgsNotation *notation,
                       AgsNote *note,
                       gboolean use_selection_list);
```

Adds *note* to *notation* .

Parameters

notation	the AgsNotation	
note	the AgsNote to add	
use_selection_list	if TRUE add to selection, else to default notation	

Since: 3.0.0

ags_notation_remove_note ()

```
void
ags_notation_remove_note (AgsNotation *notation,
                          AgsNote *note,
                          gboolean use_selection_list);
```

Removes *note* from *notation* .

Parameters

notation	the AgsNotation	
note	the AgsNote to remove	
use_selection_list	if TRUE remove from selection, else from default notation	

Since: 3.0.0

ags_notation_remove_note_at_position ()

```
gboolean
ags_notation_remove_note_at_position (AgsNotation *notation,
                                      guint x,
                                      guint y);
```

Removes one AgsNote of notation.

Parameters

notation	the AgsNotation	
x	offset	
y	note	

Returns

TRUE if successfully removed note.

Since: 3.0.0

ags_notation_get_selection ()

```
GList~*
ags_notation_get_selection (AgsNotation *notation);
```

Retrieve selection.

Parameters

notation	the AgsNotation

Returns

the selection.

[element-type AgsAudio.Note][transfer none]

Since: 3.0.0

ags_notation_is_note_selected ()

```
gboolean
ags_notation_is_note_selected (AgsNotation *notation,
                               AgsNote *note);
```

Check selection for note.

Parameters

notation	the AgsNotation
note	the AgsNote to check for

Returns

TRUE if selected otherwise FALSE

Since: 3.0.0

ags_notation_find_point ()

```
AgsNote~*
ags_notation_find_point (AgsNotation *notation,
                         guint x,
                         guint y,
                         gboolean use_selection_list);
```

Find note by offset and tone.

Parameters

notation	the AgsNotation	
x	offset	
y	note	
use_selection_list	if TRUE selection is searched	

Returns

the matching note.

[transfer none]

Since: 3.0.0

ags_notation_find_region ()

```
GList~*
ags_notation_find_region (AgsNotation *notation,
                          guint x0,
                          guint y0,
                          guint x1,
                          guint y1,
                          gboolean use_selection_list);
```

Find note by offset and tone region.

Parameters

notation	the AgsNotation	
x0	start offset	
y0	start tone	
x1	end offset	
y1	end tone	
use_selection_list	if TRUE selection is searched	

Returns

the matching notes as GList.

[element-type AgsAudio.Note][transfer container]

Since: 3.0.0

ags_notation_find_offset ()

```
GList~*
ags_notation_find_offset (AgsNotation *notation,
                          guint x,
                          gboolean use_selection_list);
```

Find all notes by offset *x* .

Parameters

notation	the AgsNotation	
x	offset	
use_selection_list	if TRUE selection is searched	

Returns

the GList containing matching AgsNote.

[element-type AgsAudio.Note][transfer full]

Since: 3.0.0

ags_notation_free_selection ()

```
void
ags_notation_free_selection (AgsNotation *notation);
```

Clear selection.

Parameters

notation	the AgsNotation	

Since: 3.0.0

ags_notation_free_all_selection ()

```
void
ags_notation_free_all_selection (GList *notation);
```

Clear all selection of `notation`.

Parameters

notation	the GList containing AgsNotation.	*[element-type AgsAudio.Notation]*

Since: 3.14.10

ags_notation_add_point_to_selection ()

```
void
ags_notation_add_point_to_selection (AgsNotation *notation,
                                     guint x,
                                     guint y,
                                     gboolean replace_current_selection);
```

Select notes at position.

Parameters

notation	the AgsNotation	
x	x offset	
y	y note tone	
replace_current_selection	if TRUE selection is replaced	

Since: 3.0.0

ags_notation_remove_point_from_selection ()

```
void
ags_notation_remove_point_from_selection
                                (AgsNotation *notation,
                                 guint x,
                                 guint y);
```

Remove notes at position of selection.

Parameters

notation	the AgsNotation	
x	x offset	
y	y note tone	

Since: 3.0.0

ags_notation_add_region_to_selection ()

```
void
ags_notation_add_region_to_selection (AgsNotation *notation,
                                      guint x0,
                                      guint y0,
                                      guint x1,
                                      guint y1,
                                      gboolean replace_current_selection);
```

Add note within region to selection.

Parameters

notation	the AgsNotation	
x0	x start offset	
y0	y start tone	
x1	x end offset	
y1	y end tone	
replace_current_selection	if TRUE selection is replaced	

Since: 3.0.0

ags_notation_remove_region_from_selection ()

```
void
ags_notation_remove_region_from_selection
                                (AgsNotation *notation,
                                 guint x0,
                                 guint y0,
                                 guint x1,
                                 guint y1);
```

Remove note within region of selection.

Parameters

notation	the AgsNotation	
x0	x start offset	
y0	y start tone	
x1	x end offset	
y1	y end tone	

Since: 3.0.0

ags_notation_add_all_to_selection ()

```
void
ags_notation_add_all_to_selection (AgsNotation *notation);
```

Add all note to selection.

Parameters

notation	the AgsNotation	

Since: 3.0.0

ags_notation_copy_selection ()

```
xmlNode~*
ags_notation_copy_selection (AgsNotation *notation);
```

Copy selection to clipboard.

Parameters

notation	the AgsNotation	

Returns

the selection as XML.

[transfer none]

Since: 3.0.0

ags_notation_cut_selection ()

```
xmlNode~*
ags_notation_cut_selection (AgsNotation *notation);
```

Cut selection to clipboard.

Parameters

notation	the AgsNotation

Returns

the selection as xmlNode.

[transfer none]

Since: 3.0.0

ags_notation_insert_from_clipboard ()

```
void
ags_notation_insert_from_clipboard (AgsNotation *notation,
                                    xmlNode *notation_node,
                                    gboolean reset_x_offset,
                                    guint x_offset,
                                    gboolean reset_y_offset,
                                    guint y_offset);
```

Paste previously copied notes.

Parameters

notation	the AgsNotation	
notation_node	the clipboard XML data	
reset_x_offset	if TRUE *x_offset* used as cursor	
x_offset	region start cursor offset	
reset_y_offset	if TRUE *y_offset* used as cursor	
y_offset	region start cursor tone	

Since: 3.0.0

ags_notation_insert_from_clipboard_extended ()

```
void
ags_notation_insert_from_clipboard_extended
                                (AgsNotation *notation,
                                 xmlNode *notation_node,
                                 gboolean reset_x_offset,
                                 guint x_offset,
                                 gboolean reset_y_offset,
                                 guint y_offset,
                                 gboolean match_channel,
                                 gboolean no_duplicates);
```

Paste previously copied notes.

Parameters

notation	the AgsNotation	
notation_node	the clipboard XML data	
reset_x_offset	if TRUE `x_offset` used as cursor	
x_offset	region start cursor offset	
reset_y_offset	if TRUE `y_offset` used as cursor	
y_offset	region start cursor tone	
match_channel	only paste if channel matches	
no_duplicates	only paste if current note doesn't exist	

Since: 3.0.0

ags_notation_to_raw_midi ()

```
guchar~*
ags_notation_to_raw_midi (AgsNotation *notation,
                          gdouble bpm,
                          gdouble delay_factor,
                          glong nn,
                          glong dd,
                          glong cc,
                          glong bb,
                          glong tempo,
                          guint *buffer_length);
```

Convert `notation` to raw-midi.

Parameters

notation	the AgsNotation	
bpm	the source bpm	
delay_factor	the source delay factor	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	
buffer_length	the return location of buffer length	

Returns

the raw-midi buffer

Since: 3.0.0

ags_notation_from_raw_midi ()

```
AgsNotation~*
ags_notation_from_raw_midi (guchar *raw_midi,
                            glong nn,
                            glong dd,
                            glong cc,
                            glong bb,
                            glong tempo,
                            gdouble bpm,
                            gdouble delay_factor,
                            guint buffer_length);
```

Parse *raw_midi* data and convert to AgsNotation.

Parameters

raw_midi	the data array	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
buffer_length	the buffer length	

Returns

the AgsNotation.

[transfer full]

Since: 3.0.0

ags_notation_new ()

```
AgsNotation~*
ags_notation_new (GObject *audio,
                  guint audio_channel);
```

Creates a new instance of AgsNotation.

Parameters

audio	the assigned AgsAudio	
audio_channel	the audio channel to apply	

Returns

the new AgsNotation

Since: 3.0.0

AGS_IS_NOTATION()

```
#define AGS_IS_NOTATION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_NOTATION))
```

AGS_IS_NOTATION_CLASS()

```
#define AGS_IS_NOTATION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_NOTATION))
```

AGS_NOTATION()

```
#define AGS_NOTATION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_NOTATION, AgsNotation))
```

AGS_NOTATION_CLASS()

```
#define AGS_NOTATION_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_NOTATION, AgsNotationClass))
```

AGS_NOTATION_GET_CLASS()

```
#define AGS_NOTATION_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_NOTATION, AgsNotationClass))
```

ags_notation_flags_get_type ()

```
GType
ags_notation_flags_get_type ();
```

ags_notation_get_type ()

```
GType
ags_notation_get_type ();
```

Types and Values

AGS_NOTATION_DEFAULT_BPM

```
#define AGS_NOTATION_DEFAULT_BPM (120.0)
```

AGS_NOTATION_TICS_PER_BEAT

```
#define AGS_NOTATION_TICS_PER_BEAT (1.0)
```

AGS_NOTATION_MINIMUM_NOTE_LENGTH

```
#define AGS_NOTATION_MINIMUM_NOTE_LENGTH (1.0 / 16.0)
```

AGS_NOTATION_MAXIMUM_NOTE_LENGTH

```
#define AGS_NOTATION_MAXIMUM_NOTE_LENGTH (16.0)
```

AGS_NOTATION_DEFAULT_LENGTH

```
#define AGS_NOTATION_DEFAULT_LENGTH (16 * 16 * 1200 / AGS_NOTATION_TICS_PER_BEAT)
```

AGS_NOTATION_DEFAULT_JIFFIE

```
#define AGS_NOTATION_DEFAULT_JIFFIE (60.0 / AGS_NOTATION_DEFAULT_BPM /  ↩
    AGS_NOTATION_TICS_PER_BEAT)
```

AGS_NOTATION_DEFAULT_DURATION

```
#define AGS_NOTATION_DEFAULT_DURATION (AGS_NOTATION_DEFAULT_LENGTH *  ↩
    AGS_NOTATION_DEFAULT_JIFFIE * AGS_USEC_PER_SEC)
```

AGS_NOTATION_DEFAULT_OFFSET

```
#define AGS_NOTATION_DEFAULT_OFFSET (64 * (1 / AGS_NOTATION_MINIMUM_NOTE_LENGTH))
```

AGS_NOTATION_DEFAULT_END

```
#define AGS_NOTATION_DEFAULT_END (64 * 64 * 1200)
```

AGS_NOTATION_DEFAULT_DIVISION

```
#define AGS_NOTATION_DEFAULT_DIVISION (96)
```

AGS_NOTATION_CLIPBOARD_VERSION

```
#define AGS_NOTATION_CLIPBOARD_VERSION "1.2.0"
```

AGS_NOTATION_CLIPBOARD_TYPE

```
#define AGS_NOTATION_CLIPBOARD_TYPE "AgsNotationClipboardXml"
```

AGS_NOTATION_CLIPBOARD_FORMAT

```
#define AGS_NOTATION_CLIPBOARD_FORMAT "AgsNotationNativePiano"
```

enum AgsNotationFlags

Enum values to control the behavior or indicate internal state of AgsNotation by enable/disable as flags.

Members

AGS_NOTATION_PATTERN_MODE	pattern mode
AGS_NOTATION_BYPASS	ignore any notation data

AGS_TYPE_NOTATION

```
#define AGS_TYPE_NOTATION                (ags_notation_get_type())
```

struct AgsNotation

```
struct AgsNotation;
```

struct AgsNotationClass

```
struct AgsNotationClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsNotation

Flags: Read / Write

Since: 3.0.0

The "audio-channel" property

```
"audio-channel"            guint
```

The effect's audio-channel.

Owner: AgsNotation

Flags: Read / Write

Allowed values: <= 65535

Default value: 0

Since: 3.0.0

The "note" property

```
"note"                     gpointer
```

The assigned AgsNote

[transfer full]

Owner: AgsNotation

Flags: Read / Write

Since: 3.0.0

The "timestamp" property

```
"timestamp"                AgsTimestamp~*
```

The notation's timestamp.

Owner: AgsNotation

Flags: Read / Write

Since: 3.0.0

10.2 AgsNote

AgsNote — Note class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_NOTE_GET_OBJ_MUTEX()
GRecMutex *	ags_note_get_obj_mutex ()
gboolean	ags_note_test_flags ()
void	ags_note_set_flags ()

void	ags_note_unset_flags ()
gint	ags_note_sort_func ()
gboolean	ags_note_get_is_minor ()
void	ags_note_set_is_minor ()
guint	ags_note_get_sharp_flats ()
void	ags_note_set_sharp_flats ()
guint	ags_note_get_x0 ()
void	ags_note_set_x0 ()
guint	ags_note_get_x1 ()
void	ags_note_set_x1 ()
guint	ags_note_get_y ()
void	ags_note_set_y ()
guint64	ags_note_get_rt_offset ()
void	ags_note_set_rt_offset ()
guint	ags_note_get_rt_attack ()
void	ags_note_set_rt_attack ()
AgsComplex *	ags_note_get_attack ()
void	ags_note_set_attack ()
AgsComplex *	ags_note_get_sustain ()
void	ags_note_set_sustain ()
AgsComplex *	ags_note_get_decay ()
void	ags_note_set_decay ()
AgsComplex *	ags_note_get_release ()
void	ags_note_set_release ()
AgsComplex *	ags_note_get_ratio ()
void	ags_note_set_ratio ()
GList *	ags_note_find_prev ()
GList *	ags_note_find_next ()
glong	ags_note_length_to_smf_delta_time ()
guint	ags_note_smf_delta_time_to_length ()
guchar *	ags_note_to_raw_midi ()
guchar *	ags_note_to_raw_midi_extended ()
snd_seq_event_t *	ags_note_to_seq_event ()
snd_seq_event_t *	ags_note_to_seq_event_extended ()
GList *	ags_note_from_raw_midi ()
GList *	ags_note_from_raw_midi_extended ()
GList *	ags_note_from_seq_event ()
GList *	ags_note_from_seq_event_extended ()
AgsNote *	ags_note_duplicate ()
AgsNote *	ags_note_new ()
AgsNote *	ags_note_new_with_offset ()
#define	AGS_IS_NOTE()
#define	AGS_IS_NOTE_CLASS()
#define	AGS_NOTE()
#define	AGS_NOTE_CLASS()
#define	AGS_NOTE_GET_CLASS()
GType	ags_note_get_type ()

Properties

AgsComplex *	attack	Read / Write
AgsComplex *	decay	Read / Write
double	frequency	Read / Write
char *	note-name	Read / Write
AgsComplex *	ratio	Read / Write

AgsComplex *	release	Read / Write
guint	rt-attack	Read / Write
guint64	rt-offset	Read / Write
double	stream-attack	Read / Write
double	stream-delay	Read / Write
guint64	stream-frame-count	Read / Write
AgsComplex *	sustain	Read / Write
guint	x0	Read / Write
guint	x1	Read / Write
guint	y	Read / Write

Types and Values

enum	AgsNoteFlags
#define	AGS_TYPE_NOTE
struct	AgsNote
struct	AgsNoteClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsNote
```

Includes

```
#include <ags/audio/ags_note.h>
```

Description

AgsNote represents a tone. The note object contains envelope information by specifying `attack`, `decay`, `sustain`, `release` and `ratio` properties as AgsComplex data type.

The real field of AgsComplex is the magnitude of the envelope and the imag field the relative offset. Raised by `ratio` real field amount.

Functions

AGS_NOTE_GET_OBJ_MUTEX()

```
#define AGS_NOTE_GET_OBJ_MUTEX(obj) (&(((AgsNote *) obj)->obj_mutex))
```

ags_note_get_obj_mutex ()

```
GRecMutex~*
ags_note_get_obj_mutex (AgsNote *note);
```

Get object mutex.

Parameters

note	the AgsNote	

Returns

the GRecMutex to lock *note*

Since: 3.1.0

ags_note_test_flags ()

```
gboolean
ags_note_test_flags (AgsNote *note,
                     guint flags);
```

Test *flags* to be set on *note* .

Parameters

note	the AgsNote	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_note_set_flags ()

```
void
ags_note_set_flags (AgsNote *note,
                    guint flags);
```

Set *flags* on *note* .

Parameters

note	the AgsNote	
flags	the flags	

Since: 3.0.0

ags_note_unset_flags ()

```
void
ags_note_unset_flags (AgsNote *note,
                      guint flags);
```

Unset *flags* on *note* .

Parameters

note	the AgsNote	
flags	the flags	

Since: 3.0.0

ags_note_sort_func ()

```
gint
ags_note_sort_func (gconstpointer a,
                    gconstpointer b);
```

Sort notes.

Parameters

a	the AgsNote	
b	another AgsNote	

Returns

0 if equal, -1 if smaller and 1 if bigger offset

Since: 3.0.0

ags_note_get_is_minor ()

```
gboolean
ags_note_get_is_minor (AgsNote *note);
```

Gets is minor.

Parameters

note	the AgsNote	

Returns

is minor

Since: 3.1.0

ags_note_set_is_minor ()

```
void
ags_note_set_is_minor (AgsNote *note,
                       gboolean is_minor);
```

Sets is minor.

Parameters

note	the AgsNote	
is_minor	is minor	

Since: 3.1.0

ags_note_get_sharp_flats ()

```
guint
ags_note_get_sharp_flats (AgsNote *note);
```

Gets sharp flats.

Parameters

note	the AgsNote	

Returns

the sharp flats

Since: 3.1.0

ags_note_set_sharp_flats ()

```
void
ags_note_set_sharp_flats (AgsNote *note,
                          guint sharp_flats);
```

Sets sharp flats.

Parameters

note	the AgsNote	
sharp_flats	the sharp flats	

Since: 3.1.0

ags_note_get_x0 ()

```
guint
ags_note_get_x0 (AgsNote *note);
```

Gets x0.

Parameters

note	the AgsNote	

Returns

the sharp flats

Since: 3.1.0

ags_note_set_x0 ()

```
void
ags_note_set_x0 (AgsNote *note,
                 guint x0);
```

Sets x0.

Parameters

note	the AgsNote	
x0	the x0	

Since: 3.1.0

ags_note_get_x1 ()

```
guint
ags_note_get_x1 (AgsNote *note);
```

Gets x1.

Parameters

note	the AgsNote	

Returns

the x1

Since: 3.1.0

ags_note_set_x1 ()

```
void
ags_note_set_x1 (AgsNote *note,
                 guint x1);
```

Sets x1.

Parameters

note	the AgsNote	
x1	the x1	

Since: 3.1.0

ags_note_get_y ()

```
guint
ags_note_get_y (AgsNote *note);
```

Gets y.

Parameters

note	the AgsNote

Returns

the y

Since: 3.1.0

ags_note_set_y ()

```
void
ags_note_set_y (AgsNote *note,
                guint y);
```

Sets y.

Parameters

note	the AgsNote
y	the y

Since: 3.1.0

ags_note_get_rt_offset ()

```
guint64
ags_note_get_rt_offset (AgsNote *note);
```

Gets rt-offset.

Parameters

note	the AgsNote

Returns

the rt-offset

Since: 3.1.0

ags_note_set_rt_offset ()

```
void
ags_note_set_rt_offset (AgsNote *note,
                        guint64 rt_offset);
```

Sets rt-offset.

Parameters

note	the AgsNote	
rt_offset	the rt-offset	

Since: 3.1.0

ags_note_get_rt_attack ()

```
guint
ags_note_get_rt_attack (AgsNote *note);
```

Gets rt-attack.

Parameters

note	the AgsNote	

Returns

the rt-attack

Since: 3.1.0

ags_note_set_rt_attack ()

```
void
ags_note_set_rt_attack (AgsNote *note,
                        guint rt_attack);
```

Sets rt-attack.

Parameters

note	the AgsNote	
rt_attack	the rt-attack	

Since: 3.1.0

ags_note_get_attack ()

```
AgsComplex~*
ags_note_get_attack (AgsNote *note);
```

Gets attack.

Parameters

note	the AgsNote

Returns

the attack

Since: 3.1.0

ags_note_set_attack ()

```
void
ags_note_set_attack (AgsNote *note,
                     AgsComplex *attack);
```

Sets attack.

Parameters

note	the AgsNote
attack	the attack

Since: 3.1.0

ags_note_get_sustain ()

```
AgsComplex~*
ags_note_get_sustain (AgsNote *note);
```

Gets sustain.

Parameters

note	the AgsNote

Returns

the sustain

Since: 3.1.0

ags_note_set_sustain ()

```
void
ags_note_set_sustain (AgsNote *note,
                      AgsComplex *sustain);
```

Sets sustain.

Parameters

note	the AgsNote	
sustain	the sustain	

Since: 3.1.0

ags_note_get_decay ()

```
AgsComplex~*
ags_note_get_decay (AgsNote *note);
```

Gets decay.

Parameters

note	the AgsNote	

Returns

the decay

Since: 3.1.0

ags_note_set_decay ()

```
void
ags_note_set_decay (AgsNote *note,
                    AgsComplex *decay);
```

Sets decay.

Parameters

note	the AgsNote	
decay	the decay	

Since: 3.1.0

ags_note_get_release ()

```
AgsComplex~*
ags_note_get_release (AgsNote *note);
```

Gets release.

Parameters

note	the AgsNote	

Returns

the release

Since: 3.1.0

ags_note_set_release ()

```
void
ags_note_set_release (AgsNote *note,
                      AgsComplex *release);
```

Sets release.

Parameters

note	the AgsNote	
release	the release	

Since: 3.1.0

ags_note_get_ratio ()

```
AgsComplex~*
ags_note_get_ratio (AgsNote *note);
```

Gets ratio.

Parameters

note	the AgsNote	

Returns

the ratio

Since: 3.1.0

ags_note_set_ratio ()

```
void
ags_note_set_ratio (AgsNote *note,
                    AgsComplex *ratio);
```

Sets ratio.

Parameters

note	the AgsNote	
ratio	the ratio	

Since: 3.1.0

ags_note_find_prev ()

```
GList~*
ags_note_find_prev (GList *note,
                    guint x0,
                    guint y);
```

Find prev note having the same y offset.

Parameters

note	the GList containing AgsNote.	*[element-type AgsAudio.Note][transfer none]*
x0	x offset	
y	y offset	

Returns

the matching entry as GList if first entry's x offset bigger than `x0` , else NULL.

[element-type AgsAudio.Note][transfer none]

Since: 3.0.0

ags_note_find_next ()

```
GList~*
ags_note_find_next (GList *note,
                    guint x0,
                    guint y);
```

Find next note having the same y offset.

Parameters

note	the GList containing AgsNote.	*[element-type AgsAudio.Note][transfer none]*
x0	x offset	
y	y offset	

Returns

the matching entry as GList if last entry's x offset smaller than `x0` , else NULL.

[element-type AgsAudio.Note][transfer none]

Since: 3.0.0

ags_note_length_to_smf_delta_time ()

```
glong
ags_note_length_to_smf_delta_time (guint note_length,
                                   gdouble bpm,
                                   gdouble delay_factor,
                                   glong nn,
                                   glong dd,
                                   glong cc,
                                   glong bb,
                                   glong tempo);
```

Convert note length to SMF delta-time.

Parameters

note_length	the note length to convert	
bpm	the source bpm	
delay_factor	the source delay factor	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	

Returns

the delta-time

Since: 3.0.0

ags_note_smf_delta_time_to_length ()

```
guint
ags_note_smf_delta_time_to_length (glong delta_time,
                                   glong nn,
                                   glong dd,
                                   glong cc,
                                   glong bb,
                                   glong tempo,
                                   gdouble bpm,
                                   gdouble delay_factor);
```

Convert SMF delta-time to note length.

Parameters

delta_time	delta-time	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	
bpm	the target bpm	
delay_factor	the target delay factor	

Returns

the note length

Since: 3.0.0

ags_note_to_raw_midi ()

```
guchar~*
ags_note_to_raw_midi (AgsNote *note,
                      gdouble bpm,
                      gdouble delay_factor,
                      guint *buffer_length);
```

Convert *note* to raw MIDI and set the buffer length of returned bytes in the array as *buffer_length* .

Parameters

note	the AgsNote	
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
buffer_length	the length of the returned buffer	

Returns

The sequencer raw midi as array.

Since: 3.0.0

ags_note_to_raw_midi_extended ()

```
guchar~*
ags_note_to_raw_midi_extended (AgsNote *note,
                               gdouble bpm,
                               gdouble delay_factor,
                               glong nn,
                               glong dd,
                               glong cc,
                               glong bb,
                               glong tempo,
                               guint *buffer_length);
```

Convert *note* to raw-midi.

Parameters

note	the AgsNote	
bpm	the source bpm	
delay_factor	the source delay factor	
nn	numerator	
dd	denominator	
cc	clocks	

bb	beats	
tempo	tempo	
buffer_length	the return location of buffer length	

Returns

the raw-midi buffer

Since: 3.0.0

ags_note_to_seq_event ()

```
snd_seq_event_t~*
ags_note_to_seq_event (AgsNote *note,
                       gdouble bpm,
                       gdouble delay_factor,
                       guint *n_events);
```

Convert *note* to ALSA sequencer events and set the number of events in the array as *n_events* .

Parameters

note	the AgsNote	
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
n_events	the count of events	

Returns

The sequencer events as array.

[type gpointer][transfer none]

Since: 3.0.0

ags_note_to_seq_event_extended ()

```
snd_seq_event_t~*
ags_note_to_seq_event_extended (AgsNote *note,
                                gdouble bpm,
                                gdouble delay_factor,
                                glong nn,
                                glong dd,
                                glong cc,
                                glong bb,
                                glong tempo,
                                guint *n_events);
```

Convert *note* to raw-midi.

Parameters

note	the AgsNote	
bpm	the source bpm	
delay_factor	the source delay factor	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	
n_events	the return location of event count	

Returns

an array of snd_seq_event_t structs.

[type gpointer][transfer none]

Since: 3.0.0

ags_note_from_raw_midi ()

```
GList~*
ags_note_from_raw_midi (guchar *raw_midi,
                        gdouble bpm,
                        gdouble delay_factor,
                        guint length);
```

Parse `raw_midi` data and convert to AgsNote.

Parameters

raw_midi	the data array	
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
length	the length of the array	

Returns

a GList containing the notes.

[element-type AgsAudio.Note][transfer full]

Since: 3.0.0

ags_note_from_raw_midi_extended ()

```
GList~*
ags_note_from_raw_midi_extended (guchar *raw_midi,
                                 glong nn,
                                 glong dd,
                                 glong cc,
                                 glong bb,
                                 glong tempo,
                                 gdouble bpm,
                                 gdouble delay_factor,
                                 guint length);
```

Parse `raw_midi` data and convert to AgsNote.

Parameters

raw_midi	the data array	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
length	the length of the array	

Returns

a GList containing the notes.

[element-type AgsAudio.Note][transfer full]

Since: 3.0.0

ags_note_from_seq_event ()

```
GList~*
ags_note_from_seq_event (snd_seq_event_t *event,
                         gdouble bpm,
                         gdouble delay_factor,
                         guint n_events);
```

Convert ALSA sequencer data `event` to AgsNote.

Parameters

event	ALSA sequencer events as array.	*[type gpointer][transfer none]*
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
n_events	the arrays length	

Returns

a GList containing the notes.

[element-type AgsAudio.Note][transfer full]

Since: 3.0.0

ags_note_from_seq_event_extended ()

```
GList~*
ags_note_from_seq_event_extended (snd_seq_event_t *event,
                                  glong nn,
                                  glong dd,
                                  glong cc,
                                  glong bb,
```

```
                                    glong tempo,
                                    gdouble bpm,
                                    gdouble delay_factor,
                                    guint n_events);
```

Parse `raw_midi` data and convert to AgsNote.

Parameters

event	the snd_seq_event_t struct array.	*[type gpointer][transfer none]*
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	
tempo	tempo	
bpm	the bpm to use	
delay_factor	the segmentation delay factor	
n_events	the count snd_seq_event_t structs	

Returns

a GList containing the notes.

[element-type AgsAudio.Note][transfer full]

Since: 3.0.0

ags_note_duplicate ()

```
AgsNote~*
ags_note_duplicate (AgsNote *note);
```

Duplicate a note.

Parameters

note	the AgsNote	

Returns

the duplicated AgsNote.

[transfer full]

Since: 3.0.0

ags_note_new ()

```
AgsNote~*
ags_note_new ();
```

Creates a new instance of AgsNote

Returns

the new AgsNote

Since: 3.0.0

ags_note_new_with_offset ()

```
AgsNote~*
ags_note_new_with_offset (guint x0,
                          guint x1,
                          guint y,
                          gdouble stream_delay,
                          gdouble stream_attack);
```

Creates a new instance of AgsNote

Parameters

x0	x0	
x1	x1	
y	y	
stream_delay	delay	
stream_attack	attack	

Returns

the new AgsNote

Since: 3.0.0

AGS_IS_NOTE()

```
#define AGS_IS_NOTE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_NOTE))
```

AGS_IS_NOTE_CLASS()

```
#define AGS_IS_NOTE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_NOTE))
```

AGS_NOTE()

```
#define AGS_NOTE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_NOTE,  ↩
    AgsNote))
```

AGS_NOTE_CLASS()

```
#define AGS_NOTE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_NOTE,  ↩
    AgsNoteClass))
```

AGS_NOTE_GET_CLASS()

```
#define AGS_NOTE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_NOTE,  ↩
    AgsNoteClass))
```

ags_note_get_type ()

```
GType
ags_note_get_type ();
```

Types and Values

enum AgsNoteFlags

Enum values to control the behavior or indicate internal state of AgsNote by enable/disable as flags.

Members

AGS_NOTE_GUI	gui format
AGS_NOTE_RUNTIME	runtime format
AGS_NOTE_HUMAN_READABLE	human readable format
AGS_NOTE_DEFAULT_LENGTH	default length
AGS_NOTE_IS_SELECTED	is selected
AGS_NOTE_FEED	feed note
AGS_NOTE_ENVELOPE	do envelope

AGS_TYPE_NOTE

```
#define AGS_TYPE_NOTE                 (ags_note_get_type())
```

struct AgsNote

```
struct AgsNote;
```

struct AgsNoteClass

```
struct AgsNoteClass {
  GObjectClass gobject;
};
```

Property Details

The "attack" property

```
  "attack"                    AgsComplex~*
```

Envelope attack.

Owner: AgsNote

Flags: Read / Write

Since: 3.0.0

The "decay" property

```
  "decay"                     AgsComplex~*
```

Envelope decay.

Owner: AgsNote

Flags: Read / Write

Since: 3.0.0

The "frequency" property

```
  "frequency"                 double
```

The note's frequency.

Owner: AgsNote

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "note-name" property

```
  "note-name"                 char~*
```

The note's name.

Owner: AgsNote

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ratio" property

```
"ratio"                    AgsComplex~*
```

Envelope ratio.

Owner: AgsNote

Flags: Read / Write

Since: 3.0.0

The "release" property

```
"release"                  AgsComplex~*
```

Envelope release.

Owner: AgsNote

Flags: Read / Write

Since: 3.0.0

The "rt-attack" property

```
"rt-attack"                guint
```

Note realtime attack.

Owner: AgsNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "rt-offset" property

```
"rt-offset"                guint64
```

Note realtime offset.

Owner: AgsNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "stream-attack" property

```
"stream-attack"            double
```

The stream's attack.

Owner: AgsNote

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "stream-delay" property

```
"stream-delay"              double
```

The stream's delay.

Owner: AgsNote

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "stream-frame-count" property

```
"stream-frame-count"        guint64
```

The stream's frame count.

Owner: AgsNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "sustain" property

```
"sustain"                   AgsComplex~*
```

Envelope sustain.

Owner: AgsNote

Flags: Read / Write

Since: 3.0.0

The "x0" property

```
"x0"                        guint
```

Note offset x0.

Owner: AgsNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "x1" property

```
"x1"                        guint
```

Note offset x1.

Owner: AgsNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "y" property

```
"y"                         guint
```

Note offset y.

Owner: AgsNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

10.3 AgsAutomation

AgsAutomation — Automation class supporting selection and clipboard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUTOMATION_GET_OBJ_MUTEX()
GRecMutex *	ags_automation_get_obj_mutex ()
gboolean	ags_automation_test_flags ()
void	ags_automation_set_flags ()
void	ags_automation_unset_flags ()
GList *	ags_automation_find_port ()
GList *	ags_automation_find_near_timestamp ()
GList *	ags_automation_find_near_timestamp_extended ()
gint	ags_automation_sort_func ()
GList *	ags_automation_add ()
GObject *	ags_automation_get_audio ()
void	ags_automation_set_audio ()
GType	ags_automation_get_channel_type ()
void	ags_automation_set_channel_type ()
guint	ags_automation_get_line ()
void	ags_automation_set_line ()
AgsTimestamp *	ags_automation_get_timestamp ()
void	ags_automation_set_timestamp ()
gchar *	ags_automation_get_control_name ()
void	ags_automation_set_control_name ()
guint	ags_automation_get_steps ()
void	ags_automation_set_steps ()
gdouble	ags_automation_get_upper ()
void	ags_automation_set_upper ()
gdouble	ags_automation_get_lower ()
void	ags_automation_set_lower ()
gdouble	ags_automation_get_default_value ()
void	ags_automation_set_default_value ()
GObject *	ags_automation_get_port ()
void	ags_automation_set_port ()

GList *	ags_automation_get_acceleration ()
void	ags_automation_set_acceleration ()
void	ags_automation_add_acceleration ()
void	ags_automation_remove_acceleration ()
gboolean	ags_automation_remove_acceleration_at_position ()
GList *	ags_automation_get_selection ()
gboolean	ags_automation_is_acceleration_selected ()
AgsAcceleration *	ags_automation_find_point ()
GList *	ags_automation_find_region ()
void	ags_automation_free_selection ()
void	ags_automation_free_all_selection ()
void	ags_automation_add_point_to_selection ()
void	ags_automation_remove_point_from_selection ()
void	ags_automation_add_region_to_selection ()
void	ags_automation_remove_region_from_selection ()
void	ags_automation_add_all_to_selection ()
xmlNode *	ags_automation_copy_selection ()
xmlNode *	ags_automation_cut_selection ()
void	ags_automation_insert_from_clipboard ()
void	ags_automation_insert_from_clipboard_extended ()
gchar **	ags_automation_get_specifier_unique ()
gchar **	ags_automation_get_specifier_unique_with_channel_type ()
GList *	ags_automation_find_specifier ()
GList *	ags_automation_find_channel_type_with_control_name ()
GList *	ags_automation_find_specifier_with_type_and_line ()
guint	ags_automation_get_value ()
AgsAutomation *	ags_automation_new ()
#define	AGS_AUTOMATION()
#define	AGS_AUTOMATION_CLASS()
#define	AGS_AUTOMATION_GET_CLASS()
#define	AGS_IS_AUTOMATION()
#define	AGS_IS_AUTOMATION_CLASS()
GType	ags_automation_flags_get_type ()
GType	ags_automation_get_type ()

Properties

gpointer	acceleration	Read / Write
AgsAudio *	audio	Read / Write
GType *	channel-type	Read / Write
char *	control-name	Read / Write
double	default-value	Read / Write
guint	line	Read / Write
double	lower	Read / Write
AgsPort *	port	Read / Write
guint	steps	Read / Write
AgsTimestamp *	timestamp	Read / Write
double	upper	Read / Write

Types and Values

#define	AGS_AUTOMATION_DEFAULT_BPM
#define	AGS_AUTOMATION_TICS_PER_BEAT
#define	AGS_AUTOMATION_MINIMUM_ACCELERATION_LENGTH

#define	AGS_AUTOMATION_MAXIMUM_ACCELERATION_LENGTH
#define	AGS_AUTOMATION_DEFAULT_LENGTH
#define	AGS_AUTOMATION_DEFAULT_JIFFIE
#define	AGS_AUTOMATION_DEFAULT_DURATION
#define	AGS_AUTOMATION_DEFAULT_OFFSET
#define	AGS_AUTOMATION_DEFAULT_PRECISION
#define	AGS_AUTOMATION_MAXIMUM_STEPS
#define	AGS_AUTOMATION_CLIPBOARD_VERSION
#define	AGS_AUTOMATION_CLIPBOARD_TYPE
#define	AGS_AUTOMATION_CLIPBOARD_FORMAT
#define	AGS_AUTOMATION_CLIPBOARD_LEGACY_FORMAT
enum	AgsAutomationFlags
#define	AGS_TYPE_AUTOMATION
struct	AgsAutomation
struct	AgsAutomationClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsAutomationFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsAutomation
```

Includes

```
#include <ags/audio/ags_automation.h>
```

Description

AgsAutomation acts as a container of AgsAcceleration. The `timestamp` property tells the engine what the first x offset of AgsAcceleration applies.

You can lookup AgsAutomation by start x offset with ags_automation_find_near_timestamp(). The next x offset is calculated as following:

next_x_offset = x_offset + AGS_AUTOMATION_DEFAULT_OFFSET;

Use ags_automation_add_acceleration() to add AgsAcceleration to AgsAutomation and ags_automation_remove_acceleration() to remove it again.

In order to copy or cut accelerations you select them first by calling ags_automation_add_region_to_selection().

Functions

AGS_AUTOMATION_GET_OBJ_MUTEX()

```
#define AGS_AUTOMATION_GET_OBJ_MUTEX(obj) (&(((AgsAutomation *) obj)->obj_mutex))
```

ags_automation_get_obj_mutex ()

```
GRecMutex~*
ags_automation_get_obj_mutex (AgsAutomation *automation);
```

Get object mutex.

Parameters

automation	the AgsAutomation

Returns

the GRecMutex to lock *automation*

Since: 3.1.0

ags_automation_test_flags ()

```
gboolean
ags_automation_test_flags (AgsAutomation *automation,
                           guint flags);
```

Test *flags* to be set on *automation* .

Parameters

automation	the AgsAutomation
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_automation_set_flags ()

```
void
ags_automation_set_flags (AgsAutomation *automation,
                          guint flags);
```

Set *flags* on *automation* .

Parameters

automation	the AgsAutomation
flags	the flags

Since: 3.0.0

ags_automation_unset_flags ()

```
void
ags_automation_unset_flags (AgsAutomation *automation,
                            guint flags);
```

Unset *flags* on *automation* .

Parameters

automation	the AgsAutomation	
flags	the flags	

Since: 3.0.0

ags_automation_find_port ()

```
GList~*
ags_automation_find_port (GList *automation,
                          GObject *port);
```

Find automation by port.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
port	the AgsPort to match	

Returns

next matching automation as GList or NULL if not found.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_find_near_timestamp ()

```
GList~*
ags_automation_find_near_timestamp (GList *automation,
                                    guint line,
                                    AgsTimestamp *timestamp);
```

Retrieve appropriate automation for timestamp.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
line	the matching audio channel	
timestamp	the matching timestamp	

Returns

Next matching GList or NULL if not found.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_find_near_timestamp_extended ()

```
GList~*
ags_automation_find_near_timestamp_extended
                                (GList *automation,
                                 guint line,
                                 GType channel_type,
                                 gchar *control_name,
                                 AgsTimestamp *timestamp);
```

Retrieve appropriate automation for timestamp.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
line	the matching audio channel	
channel_type	the matching channel type	
control_name	the matching control name	
timestamp	the matching timestamp	

Returns

Next matching GList or NULL if not found.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_sort_func ()

```
gint
ags_automation_sort_func (gconstpointer a,
                          gconstpointer b);
```

Compare *a* to *b* .

Parameters

a	an AgsAutomation	
b	an other AgsAutomation	

Returns

0 if equal, -1 if smaller and 1 if bigger offset

Since: 3.0.0

ags_automation_add ()

```
GList~*
ags_automation_add (GList *automation,
                    AgsAutomation *new_automation);
```

Add *new_automation* sorted to *automation*

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
new_automation	the AgsAutomation to add	

Returns

the new beginning of `automation`.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_get_audio ()

```
GObject~*
ags_automation_get_audio (AgsAutomation *automation);
```

Get audio.

Parameters

automation	the AgsAutomation	

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_automation_set_audio ()

```
void
ags_automation_set_audio (AgsAutomation *automation,
                          GObject *audio);
```

Set audio.

Parameters

automation	the AgsAutomation	
audio	the AgsAudio	

Since: 3.1.0

ags_automation_get_channel_type ()

```
GType
ags_automation_get_channel_type (AgsAutomation *automation);
```

Gets channel type.

Parameters

automation	the AgsAutomation

Returns

the channel type

Since: 3.1.0

ags_automation_set_channel_type ()

```
void
ags_automation_set_channel_type (AgsAutomation *automation,
                                 GType channel_type);
```

Sets channel type.

Parameters

automation	the AgsAutomation
channel_type	the channel type

Since: 3.1.0

ags_automation_get_line ()

```
guint
ags_automation_get_line (AgsAutomation *automation);
```

Gets line.

Parameters

automation	the AgsAutomation

Returns

the line

Since: 3.1.0

ags_automation_set_line ()

```
void
ags_automation_set_line (AgsAutomation *automation,
                         guint line);
```

Sets line.

Parameters

automation	the AgsAutomation	
line	the line	

Since: 3.1.0

ags_automation_get_timestamp ()

```
AgsTimestamp~*
ags_automation_get_timestamp (AgsAutomation *automation);
```

Get timestamp.

Parameters

automation	the AgsAutomation	

Returns

the AgsTimestamp.

[transfer full]

Since: 3.1.0

ags_automation_set_timestamp ()

```
void
ags_automation_set_timestamp (AgsAutomation *automation,
                              AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

automation	the AgsAutomation	
timestamp	the AgsTimestamp	

Since: 3.1.0

ags_automation_get_control_name ()

```
gchar~*
ags_automation_get_control_name (AgsAutomation *automation);
```

Get control name.

Parameters

automation	the AgsAutomation	

Returns

the control name.

[transfer full]

Since: 3.1.0

ags_automation_set_control_name ()

```
void
ags_automation_set_control_name (AgsAutomation *automation,
                                 gchar *control_name);
```

Set control name.

Parameters

automation	the AgsAutomation	
control_name	the control name	

Since: 3.1.0

ags_automation_get_steps ()

```
guint
ags_automation_get_steps (AgsAutomation *automation);
```

Gets steps.

Parameters

automation	the AgsAutomation	

Returns

the steps

Since: 3.1.0

ags_automation_set_steps ()

```
void
ags_automation_set_steps (AgsAutomation *automation,
                          guint steps);
```

Sets steps.

Parameters

automation	the AgsAutomation	
steps	the steps	

Since: 3.1.0

ags_automation_get_upper ()

```
gdouble
ags_automation_get_upper (AgsAutomation *automation);
```

Gets upper.

Parameters

automation	the AgsAutomation	

Returns

the upper

Since: 3.1.0

ags_automation_set_upper ()

```
void
ags_automation_set_upper (AgsAutomation *automation,
                          gdouble upper);
```

Sets upper.

Parameters

automation	the AgsAutomation	
upper	the upper	

Since: 3.1.0

ags_automation_get_lower ()

```
gdouble
ags_automation_get_lower (AgsAutomation *automation);
```

Gets lower.

Parameters

automation	the AgsAutomation	

Returns

the lower

Since: 3.1.0

ags_automation_set_lower ()

```
void
ags_automation_set_lower (AgsAutomation *automation,
                          gdouble lower);
```

Sets lower.

Parameters

automation	the AgsAutomation	
lower	the lower	

Since: 3.1.0

ags_automation_get_default_value ()

```
gdouble
ags_automation_get_default_value (AgsAutomation *automation);
```

Gets default value.

Parameters

automation	the AgsAutomation	

Returns

the default value

Since: 3.1.0

ags_automation_set_default_value ()

```
void
ags_automation_set_default_value (AgsAutomation *automation,
                                  gdouble default_value);
```

Sets default value.

Parameters

automation	the AgsAutomation	
default_value	the default value	

Since: 3.1.0

ags_automation_get_port ()

```
GObject~*
ags_automation_get_port (AgsAutomation *automation);
```

Get port.

Parameters

automation	the AgsAutomation

Returns

the AgsPort.

[transfer full]

Since: 3.1.0

ags_automation_set_port ()

```
void
ags_automation_set_port (AgsAutomation *automation,
                         GObject *port);
```

Set port.

Parameters

automation	the AgsAutomation
port	the AgsPort

Since: 3.1.0

ags_automation_get_acceleration ()

```
GList~*
ags_automation_get_acceleration (AgsAutomation *automation);
```

Get acceleration.

Parameters

automation	the AgsAutomation

Returns

the GList containig AgsAcceleration.

[element-type AgsAudio.Acceleration][transfer full]

Since: 3.1.0

ags_automation_set_acceleration ()

```
void
ags_automation_set_acceleration (AgsAutomation *automation,
                                 GList *acceleration);
```

Set acceleration by replacing existing.

Parameters

automation	the AgsAutomation	
acceleration	the GList containing AgsAcceleration.	*[element-type AgsAudio.Acceleration][transfer full]*

Since: 3.1.0

ags_automation_add_acceleration ()

```
void
ags_automation_add_acceleration (AgsAutomation *automation,
                                 AgsAcceleration *acceleration,
                                 gboolean use_selection_list);
```

Adds `acceleration` to `automation`.

Parameters

automation	the AgsAutomation	
acceleration	the AgsAcceleration to add	
use_selection_list	if TRUE add to selection, else to default automation	

Since: 3.0.0

ags_automation_remove_acceleration ()

```
void
ags_automation_remove_acceleration (AgsAutomation *automation,
                                    AgsAcceleration *acceleration,
                                    gboolean use_selection_list);
```

Removes `acceleration` from `automation`.

Parameters

automation	the AgsAutomation	
acceleration	the AgsAcceleration to remove	
use_selection_list	if TRUE remove from selection, else from default automation	

Since: 3.0.0

ags_automation_remove_acceleration_at_position ()

```
gboolean
ags_automation_remove_acceleration_at_position
                                (AgsAutomation *automation,
                                 guint x,
                                 gdouble y);
```

Removes one AgsAcceleration of automation.

Parameters

automation	the AgsAutomation	
x	x offset	
y	y value of acceleration	

Returns

TRUE if successfully removed acceleration, else FALSE

Since: 3.0.0

ags_automation_get_selection ()

```
GList~*
ags_automation_get_selection (AgsAutomation *automation);
```

Retrieve selection.

Parameters

automation	the AgsAutomation	

Returns

the selection.

[element-type AgsAudio.Acceleration][transfer none]

Since: 3.0.0

ags_automation_is_acceleration_selected ()

```
gboolean
ags_automation_is_acceleration_selected
                                (AgsAutomation *automation,
                                 AgsAcceleration *acceleration);
```

Check selection for acceleration.

Parameters

automation	the AgsAutomation	
acceleration	the AgsAcceleration to check for	

Returns

TRUE if selected, else FALSE

Since: 3.0.0

ags_automation_find_point ()

```
AgsAcceleration~*
ags_automation_find_point (AgsAutomation *automation,
                           guint x,
                           gdouble y,
                           gboolean use_selection_list);
```

Find acceleration by offset and acceleration.

Parameters

automation	the AgsAutomation	
x	x offset	
y	y value acceleration, will be ignored	
use_selection_list	if TRUE selection is searched	

Returns

the matching acceleration.

[transfer none]

Since: 3.0.0

ags_automation_find_region ()

```
GList~*
ags_automation_find_region (AgsAutomation *automation,
                            guint x0,
                            gdouble y0,
                            guint x1,
                            gdouble y1,
                            gboolean use_selection_list);
```

Find acceleration by offset and value region.

Parameters

automation	the AgsAutomation	
x0	start offset	
y0	value start	
x1	end offset	
y1	value end	
use_selection_list	if TRUE selection is searched	

Returns

the matching accelerations as GList.

[element-type AgsAudio.Acceleration][transfer full]

Since: 3.0.0

ags_automation_free_selection ()

```
void
ags_automation_free_selection (AgsAutomation *automation);
```

Clear selection.

Parameters

automation	the AgsAutomation	

Since: 3.0.0

ags_automation_free_all_selection ()

```
void
ags_automation_free_all_selection (GList *automation);
```

Clear all selection of `automation`.

Parameters

automation	the GList containing AgsAutomation	

Since: 3.14.10

ags_automation_add_point_to_selection ()

```
void
ags_automation_add_point_to_selection (AgsAutomation *automation,
                                       guint x,
                                       gdouble y,
                                       gboolean replace_current_selection);
```

Select acceleration at position.

Parameters

automation	the AgsAutomation	
x	x offset	
y	y acceleration value	
replace_current_selection	if TRUE selection is replaced	

Since: 3.0.0

ags_automation_remove_point_from_selection ()

```
void
ags_automation_remove_point_from_selection
                                (AgsAutomation *automation,
                                 guint x,
                                 gdouble y);
```

Remove acceleration at position of selection.

Parameters

automation	the AgsAutomation	
x	x offset	
y	y acceleration value	

Since: 3.0.0

ags_automation_add_region_to_selection ()

```
void
ags_automation_add_region_to_selection
                                (AgsAutomation *automation,
                                 guint x0,
                                 gdouble y0,
                                 guint x1,
                                 gdouble y1,
                                 gboolean replace_current_selection);
```

ags_automation_remove_region_from_selection ()

```
void
ags_automation_remove_region_from_selection
                                (AgsAutomation *automation,
                                 guint x0,
                                 gdouble y0,
                                 guint x1,
                                 gdouble y1);
```

Remove acceleration within region of selection.

Parameters

automation	the AgsAutomation	
x0	x start offset	
y0	y start acceleration	
x1	x end offset	
y1	y end acceleration	

Since: 3.0.0

ags_automation_add_all_to_selection ()

```
void
ags_automation_add_all_to_selection (AgsAutomation *automation);
```

Add all acceleration to selection.

Parameters

automation	the AgsAutomation	

Since: 3.0.0

ags_automation_copy_selection ()

```
xmlNode~*
ags_automation_copy_selection (AgsAutomation *automation);
```

Copy selection to clipboard.

Parameters

automation	the AgsAutomation	

Returns

the selection as xmlNode.

[transfer none]

Since: 3.0.0

ags_automation_cut_selection ()

```
xmlNode~*
ags_automation_cut_selection (AgsAutomation *automation);
```

Cut selection to clipboard.

Parameters

automation	the AgsAutomation	

Returns

the selection as xmlNod.

[transfer none]

Since: 3.0.0

ags_automation_insert_from_clipboard ()

```
void
ags_automation_insert_from_clipboard (AgsAutomation *automation,
                                      xmlNode *automation_node,
                                      gboolean reset_x_offset,
                                      guint x_offset,
                                      gboolean reset_y_offset,
                                      gdouble y_offset);
```

Insert clipboard `automation_node` to `automation`.

Parameters

automation	the AgsAutomation	
automation_node	the xmlNode	
reset_x_offset	if TRUE reset x offset	
x_offset	the x offset to use	
reset_y_offset	if TRUE reset y offset	
y_offset	the y offset to use	

Since: 3.0.0

ags_automation_insert_from_clipboard_extended ()

```
void
ags_automation_insert_from_clipboard_extended
                                (AgsAutomation *automation,
                                 xmlNode *automation_node,
                                 gboolean reset_x_offset,
                                 guint x_offset,
                                 gboolean reset_y_offset,
                                 gdouble y_offset,
                                 gboolean match_line,
                                 gboolean no_duplicates);
```

Insert clipboard `automation_node` to `automation`.

Parameters

automation	the AgsAutomation	
automation_node	the xmlNode	
reset_x_offset	if TRUE reset x offset	
x_offset	the x offset to use	
reset_y_offset	if TRUE reset y offset	
y_offset	the y offset to use	
match_line	if TRUE match line or discard	
no_duplicates	if TRUE eliminate duplicates	

Since: 3.0.0

ags_automation_get_specifier_unique ()

```
gchar~**
ags_automation_get_specifier_unique (GList *automation);
```

Retrieve automation port specifier.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*

Returns

a NULL terminated string array.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_automation_get_specifier_unique_with_channel_type ()

```
gchar~**
ags_automation_get_specifier_unique_with_channel_type
                                (GList *automation,
                                 GType channel_type);
```

Retrieve automation port specifier.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
channel_type	the channel's GType	

Returns

a NULL terminated string array.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_automation_find_specifier ()

```
GList~*
ags_automation_find_specifier (GList *automation,
                               gchar *specifier);
```

Find port specifier.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
specifier	the string specifier to find	

Returns

Next matching GList.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_find_channel_type_with_control_name ()

```
GList~*
ags_automation_find_channel_type_with_control_name
                               (GList *automation,
                                GType channel_type,
                                gchar *specifier);
```

Find automation by `channel_type`.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
channel_type	the GType to match	
specifier	the control name	

Returns

next matching automation as GList or NULL if not found.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_find_specifier_with_type_and_line ()

```
GList~*
ags_automation_find_specifier_with_type_and_line
                                (GList *automation,
                                 gchar *specifier,
                                 GType channel_type,
                                 guint line);
```

Find port specifier with channel type and line.

Parameters

automation	the GList containing AgsAutomation.	*[element-type AgsAudio.Automation][transfer none]*
specifier	the string specifier to find	
channel_type	the channel GType	
line	the line	

Returns

Next matching GList.

[element-type AgsAudio.Automation][transfer none]

Since: 3.0.0

ags_automation_get_value ()

```
guint
ags_automation_get_value (AgsAutomation *automation,
                          guint x,
                          guint x_end,
                          gboolean use_prev_on_failure,
                          GValue *value);
```

Get automation value.

Parameters

automation	the AgsAutomation	
x	the x-offset	
x_end	the x-end-offset	
use_prev_on_failure	if TRUE use previous value, else return G_MAXUINT	
value	the return location of value.	*[out caller-allocates]*

Returns

the x_offset

Since: 3.0.0

ags_automation_new ()

```
AgsAutomation~*
ags_automation_new (GObject *audio,
                    guint line,
                    GType channel_type,
                    gchar *control_name);
```

Creates a new instance of AgsAutomation.

Parameters

audio	the AgsAudio	
line	the line to apply	
channel_type	the channel type	
control_name	the control name	

Returns

the new AgsAutomation

Since: 3.0.0

AGS_AUTOMATION()

```
#define AGS_AUTOMATION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUTOMATION, AgsAutomation))
```

AGS_AUTOMATION_CLASS()

```
#define AGS_AUTOMATION_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_AUTOMATION, AgsAutomationClass))
```

AGS_AUTOMATION_GET_CLASS()

```
#define AGS_AUTOMATION_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_AUTOMATION, AgsAutomationClass))
```

AGS_IS_AUTOMATION()

```
#define AGS_IS_AUTOMATION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUTOMATION))
```

AGS_IS_AUTOMATION_CLASS()

```
#define AGS_IS_AUTOMATION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUTOMATION))
```

ags_automation_flags_get_type ()

```
GType
ags_automation_flags_get_type (void);
```

ags_automation_get_type ()

```
GType
ags_automation_get_type (void);
```

Types and Values

AGS_AUTOMATION_DEFAULT_BPM

```
#define AGS_AUTOMATION_DEFAULT_BPM (120.0)
```

AGS_AUTOMATION_TICS_PER_BEAT

```
#define AGS_AUTOMATION_TICS_PER_BEAT (1.0)
```

AGS_AUTOMATION_MINIMUM_ACCELERATION_LENGTH

```
#define AGS_AUTOMATION_MINIMUM_ACCELERATION_LENGTH (1.0 / 16.0 / 64.0)
```

AGS_AUTOMATION_MAXIMUM_ACCELERATION_LENGTH

```
#define AGS_AUTOMATION_MAXIMUM_ACCELERATION_LENGTH (16.0)
```

AGS_AUTOMATION_DEFAULT_LENGTH

```
#define AGS_AUTOMATION_DEFAULT_LENGTH (64 * 16 * 16 * 1200 / AGS_AUTOMATION_TICS_PER_BEAT)
```

AGS_AUTOMATION_DEFAULT_JIFFIE

```
#define AGS_AUTOMATION_DEFAULT_JIFFIE (60.0 / AGS_AUTOMATION_DEFAULT_BPM /  ↩
    AGS_AUTOMATION_TICS_PER_BEAT)
```

AGS_AUTOMATION_DEFAULT_DURATION

```
#define AGS_AUTOMATION_DEFAULT_DURATION (AGS_AUTOMATION_DEFAULT_LENGTH *  ↩
    AGS_AUTOMATION_DEFAULT_JIFFIE * AGS_USEC_PER_SEC)
```

AGS_AUTOMATION_DEFAULT_OFFSET

```
#define AGS_AUTOMATION_DEFAULT_OFFSET (64 * (1 / AGS_AUTOMATION_MINIMUM_ACCELERATION_LENGTH ↩
    ))
```

AGS_AUTOMATION_DEFAULT_PRECISION

```
#define AGS_AUTOMATION_DEFAULT_PRECISION (8)
```

AGS_AUTOMATION_MAXIMUM_STEPS

```
#define AGS_AUTOMATION_MAXIMUM_STEPS (128)
```

AGS_AUTOMATION_CLIPBOARD_VERSION

```
#define AGS_AUTOMATION_CLIPBOARD_VERSION "1.3.0"
```

AGS_AUTOMATION_CLIPBOARD_TYPE

```
#define AGS_AUTOMATION_CLIPBOARD_TYPE "AgsAutomationClipboardXml"
```

AGS_AUTOMATION_CLIPBOARD_FORMAT

```
#define AGS_AUTOMATION_CLIPBOARD_FORMAT "AgsAutomationNativeScale"
```

AGS_AUTOMATION_CLIPBOARD_LEGACY_FORMAT

```
#define AGS_AUTOMATION_CLIPBOARD_LEGACY_FORMAT "AgsAutomationNativePiano"
```

enum AgsAutomationFlags

Enum values to control the behavior or indicate internal state of AgsAutomation by enable/disable as flags.

Members

AGS_AUTOMATION_BYPASS	ignore any automation data

AGS_TYPE_AUTOMATION

```
#define AGS_TYPE_AUTOMATION                (ags_automation_get_type())
```

struct AgsAutomation

```
struct AgsAutomation;
```

struct AgsAutomationClass

```
struct AgsAutomationClass {
  GObjectClass gobject;
};
```

Property Details

The "acceleration" property

```
  "acceleration"              gpointer
```

The acceleration list.

[transfer full]

Owner: AgsAutomation

Flags: Read / Write

Since: 3.0.0

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsAutomation

Flags: Read / Write

Since: 3.0.0

The "channel-type" property

```
  "channel-type"              GType~*
```

The effect's assigned channel type.

Owner: AgsAutomation

Flags: Read / Write

Allowed values: void

Since: 3.0.0

The "control-name" property

```
"control-name"              char~*
```

The effect's assigned control name.

Owner: AgsAutomation

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "default-value" property

```
"default-value"             double
```

The effect's default-value.

Owner: AgsAutomation

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "line" property

```
"line"                      guint
```

The effect's line.

Owner: AgsAutomation

Flags: Read / Write

Allowed values: <= 65535

Default value: 0

Since: 3.0.0

The "lower" property

```
"lower"                     double
```

The effect's lower.

Owner: AgsAutomation

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "port" property

```
"port"                    AgsPort~*
```

The assigned AgsPort

Owner: AgsAutomation

Flags: Read / Write

Since: 3.0.0

The "steps" property

```
"steps"                   guint
```

The effect's steps.

Owner: AgsAutomation

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "timestamp" property

```
"timestamp"               AgsTimestamp~*
```

The automation's timestamp.

Owner: AgsAutomation

Flags: Read / Write

Since: 3.0.0

The "upper" property

```
"upper"                   double
```

The effect's upper.

Owner: AgsAutomation

Flags: Read / Write

Default value: 0

Since: 3.0.0

10.4 AgsAcceleration

AgsAcceleration — Acceleration class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_ACCELERATION_GET_OBJ_MUTEX()
GRecMutex *	ags_acceleration_get_obj_mutex ()
gboolean	ags_acceleration_test_flags ()
void	ags_acceleration_set_flags ()
void	ags_acceleration_unset_flags ()
gint	ags_acceleration_sort_func ()
guint	ags_acceleration_get_x ()
void	ags_acceleration_set_x ()
gdouble	ags_acceleration_get_y ()
void	ags_acceleration_set_y ()
AgsAcceleration *	ags_acceleration_duplicate ()
AgsAcceleration *	ags_acceleration_new ()
#define	AGS_ACCELERATION()
#define	AGS_ACCELERATION_CLASS()
#define	AGS_ACCELERATION_GET_CLASS()
#define	AGS_IS_ACCELERATION()
#define	AGS_IS_ACCELERATION_CLASS()
GType	ags_acceleration_flags_get_type ()
GType	ags_acceleration_get_type ()

Properties

char *	acceleration-name	Read / Write
guint	x	Read / Write
double	y	Read / Write

Types and Values

enum	AgsAccelerationFlags
#define	AGS_TYPE_ACCELERATION
struct	AgsAcceleration
struct	AgsAccelerationClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsAccelerationFlags
GObject
&#x2570;&#x2500;&#x2500; AgsAcceleration
```

Includes

```
#include <ags/audio/ags_acceleration.h>
```

Description

AgsAcceleration represents an automated value of a port. You specify y the value written to a port by offset x.

Functions

AGS_ACCELERATION_GET_OBJ_MUTEX()

```
#define AGS_ACCELERATION_GET_OBJ_MUTEX(obj) (&(((AgsAcceleration *) obj)->obj_mutex))
```

ags_acceleration_get_obj_mutex ()

```
GRecMutex~*
ags_acceleration_get_obj_mutex (AgsAcceleration *acceleration);
```

Get object mutex.

Parameters

acceleration	the AgsAcceleration	

Returns

the GRecMutex to lock *acceleration*

Since: 3.1.0

ags_acceleration_test_flags ()

```
gboolean
ags_acceleration_test_flags (AgsAcceleration *acceleration,
                             guint flags);
```

Test *flags* to be set on *acceleration* .

Parameters

acceleration	the AgsAcceleration	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_acceleration_set_flags ()

```
void
ags_acceleration_set_flags (AgsAcceleration *acceleration,
                            guint flags);
```

Set *flags* on *acceleration* .

Parameters

acceleration	the AgsAcceleration	
flags	the flags	

Since: 3.0.0

ags_acceleration_unset_flags ()

```
void
ags_acceleration_unset_flags (AgsAcceleration *acceleration,
                              guint flags);
```

Unset *flags* on *acceleration*.

Parameters

acceleration	the AgsAcceleration	
flags	the flags	

Since: 3.0.0

ags_acceleration_sort_func ()

```
gint
ags_acceleration_sort_func (gconstpointer a,
                            gconstpointer b);
```

Sort accelerations.

Parameters

a	an AgsAcceleration	
b	an other AgsAcceleration	

Returns

0 if equal, -1 if smaller and 1 if bigger offset

Since: 3.0.0

ags_acceleration_get_x ()

```
guint
ags_acceleration_get_x (AgsAcceleration *acceleration);
```

Gets x.

Parameters

acceleration	the AgsAcceleration

Returns

the x

Since: 3.1.0

ags_acceleration_set_x ()

```
void
ags_acceleration_set_x (AgsAcceleration *acceleration,
                        guint x);
```

Sets x.

Parameters

acceleration	the AgsAcceleration
x	the x

Since: 3.1.0

ags_acceleration_get_y ()

```
gdouble
ags_acceleration_get_y (AgsAcceleration *acceleration);
```

Gets y.

Parameters

acceleration	the AgsAcceleration

Returns

the y

Since: 3.1.0

ags_acceleration_set_y ()

```
void
ags_acceleration_set_y (AgsAcceleration *acceleration,
                        gdouble y);
```

Sets y.

Parameters

acceleration	the AgsAcceleration	
y	the y	

Since: 3.1.0

ags_acceleration_duplicate ()

```
AgsAcceleration~*
ags_acceleration_duplicate (AgsAcceleration *acceleration);
```

Duplicate a acceleration.

Parameters

acceleration	an AgsAcceleration	

Returns

the duplicated AgsAcceleration.

[transfer full]

Since: 3.0.0

ags_acceleration_new ()

```
AgsAcceleration~*
ags_acceleration_new ();
```

Creates a new instance of AgsAcceleration

Returns

the new AgsAcceleration

Since: 3.0.0

AGS_ACCELERATION()

```
#define AGS_ACCELERATION(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ACCELERATION, AgsAcceleration))
```

AGS_ACCELERATION_CLASS()

```
#define AGS_ACCELERATION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_ACCELERATION, AgsAccelerationClass))
```

AGS_ACCELERATION_GET_CLASS()

```
#define AGS_ACCELERATION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_ACCELERATION, AgsAccelerationClass))
```

AGS_IS_ACCELERATION()

```
#define AGS_IS_ACCELERATION(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_ACCELERATION))
```

AGS_IS_ACCELERATION_CLASS()

```
#define AGS_IS_ACCELERATION_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_ACCELERATION))
```

ags_acceleration_flags_get_type ()

```
GType
ags_acceleration_flags_get_type ();
```

ags_acceleration_get_type ()

```
GType
ags_acceleration_get_type ();
```

Types and Values

enum AgsAccelerationFlags

Members

AGS_ACCELERATION_DEFAULT_START	if start is default start point
AGS_ACCELERATION_DEFAULT_END	if end is default end point

AGS_ACCELERATION_GUI	interpret x and y as GUI format
AGS_ACCELERATION_RUNTIME	interpret x and y as runtime formant
AGS_ACCELERATION_HUMAN_READABLE	interpret x and y as human readable
AGS_ACCELERATION_DEFAULT_LENGTH	if default length applies
AGS_ACCELERATION_IS_SELECTED	if the acceleration is selected

AGS_TYPE_ACCELERATION

```
#define AGS_TYPE_ACCELERATION                (ags_acceleration_get_type())
```

struct AgsAcceleration

```
struct AgsAcceleration;
```

struct AgsAccelerationClass

```
struct AgsAccelerationClass {
  GObjectClass gobject;
};
```

Property Details

The "acceleration-name" property

```
  "acceleration-name"         char~*
```

The acceleration's name.

Owner: AgsAcceleration

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "x" property

```
  "x"                         guint
```

Acceleration offset x.

Owner: AgsAcceleration

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "y" property

```
  "y"                         double
```

Acceleration value y.

Owner: AgsAcceleration

Flags: Read / Write

Default value: 0

Since: 3.0.0

10.5 AgsWave

AgsWave — Wave class supporting selection and clipboard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_WAVE_GET_OBJ_MUTEX()
GRecMutex *	ags_wave_get_obj_mutex ()
gboolean	ags_wave_test_flags ()
void	ags_wave_set_flags ()
void	ags_wave_unset_flags ()
GList *	ags_wave_find_near_timestamp ()
gint	ags_wave_sort_func ()
GObject *	ags_wave_get_audio ()
void	ags_wave_set_audio ()
guint	ags_wave_get_line ()
void	ags_wave_set_line ()
guint	ags_wave_get_samplerate ()
void	ags_wave_set_samplerate ()
guint	ags_wave_get_buffer_size ()
void	ags_wave_set_buffer_size ()
guint	ags_wave_get_format ()
void	ags_wave_set_format ()
AgsTimestamp *	ags_wave_get_timestamp ()
void	ags_wave_set_timestamp ()
GList *	ags_wave_get_buffer ()
void	ags_wave_set_buffer ()
GList *	ags_wave_add ()
void	ags_wave_add_buffer ()
void	ags_wave_remove_buffer ()
GList *	ags_wave_get_selection ()
gboolean	ags_wave_is_buffer_selected ()
AgsBuffer *	ags_wave_find_point ()
AgsBuffer *	ags_wave_find_exact_point ()
GList *	ags_wave_find_region ()
void	ags_wave_free_selection ()
void	ags_wave_free_all_selection ()
void	ags_wave_add_region_to_selection ()
void	ags_wave_remove_region_from_selection ()
void	ags_wave_add_all_to_selection ()
xmlNode *	ags_wave_copy_selection ()
xmlNode *	ags_wave_cut_selection ()
void	ags_wave_insert_from_clipboard ()
void	ags_wave_insert_from_clipboard_extended ()
AgsWave *	ags_wave_new ()
#define	AGS_IS_WAVE()
#define	AGS_IS_WAVE_CLASS()
#define	AGS_WAVE()
#define	AGS_WAVE_CLASS()
#define	AGS_WAVE_GET_CLASS()
GType	ags_wave_flags_get_type ()
GType	ags_wave_get_type ()

Properties

AgsAudio *	audio	Read / Write
gpointer	buffer	Read / Write
guint	buffer-size	Read / Write
guint	format	Read / Write

guint	line	Read / Write
guint	samplerate	Read / Write
AgsTimestamp *	timestamp	Read / Write

Types and Values

#define	AGS_WAVE_DEFAULT_BPM
#define	AGS_WAVE_TICS_PER_BEAT
#define	AGS_WAVE_MINIMUM_BUFFER_LENGTH
#define	AGS_WAVE_DEFAULT_BUFFER_LENGTH
#define	AGS_WAVE_DEFAULT_LENGTH
#define	AGS_WAVE_DEFAULT_JIFFIE
#define	AGS_WAVE_DEFAULT_DURATION
#define	AGS_WAVE_DEFAULT_OFFSET
#define	AGS_WAVE_CLIPBOARD_VERSION
#define	AGS_WAVE_CLIPBOARD_TYPE
#define	AGS_WAVE_CLIPBOARD_FORMAT
enum	AgsWaveFlags
#define	AGS_TYPE_WAVE
struct	AgsWave
struct	AgsWaveClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsWaveFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsWave
```

Includes

```
#include <ags/audio/ags_wave.h>
```

Description

AgsWave acts as a container of AgsBuffer. The `timestamp` property tells the engine what the first x offset of AgsBuffer applies.

You can lookup AgsBuffer by start x offet with ags_wave_find_near_timestamp(). The next x offset is calculated as following:

next_x_offset = x_offset + relative_offset;

The relative_offset is calculated as following:

relative_offset = AGS_WAVE_DEFAULT_BUFFER_LENGTH * samplerate;

Use ags_wave_add_buffer() to add AgsBuffer to AgsWave and ags_wave_remove_buffer() to remove it again.

In order to copy or cut accelerations you select them first by calling ags_wave_add_region_to_selection().

Functions

AGS_WAVE_GET_OBJ_MUTEX()

```
#define AGS_WAVE_GET_OBJ_MUTEX(obj) (&(((AgsWave *) obj)->obj_mutex))
```

ags_wave_get_obj_mutex ()

```
GRecMutex~*
ags_wave_get_obj_mutex (AgsWave *wave);
```

Get object mutex.

Parameters

wave	the AgsWave

Returns

the GRecMutex to lock *wave*

Since: 3.1.0

ags_wave_test_flags ()

```
gboolean
ags_wave_test_flags (AgsWave *wave,
                     guint flags);
```

Test *flags* to be set on *wave* .

Parameters

wave	the AgsWave
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_wave_set_flags ()

```
void
ags_wave_set_flags (AgsWave *wave,
                    guint flags);
```

Set *flags* on *wave* .

Parameters

wave	the AgsWave
flags	the flags

Since: 3.0.0

ags_wave_unset_flags ()

```
void
ags_wave_unset_flags (AgsWave *wave,
                      guint flags);
```

Unset *flags* on *wave* .

Parameters

wave	the AgsWave	
flags	the flags	

Since: 3.0.0

ags_wave_find_near_timestamp ()

```
GList~*
ags_wave_find_near_timestamp (GList *wave,
                              guint line,
                              AgsTimestamp *timestamp);
```

Retrieve appropriate wave for timestamp.

Parameters

wave	the GList containing AgsWave.	*[element-type AgsAudio.Wave][transfer none]*
line	the matching audio channel	
timestamp	the matching AgsTimestamp, or NULL to match any timestamp	

Returns

Next matching GList or NULL if not found.

[element-type AgsAudio.Wave][transfer none]

Since: 3.0.0

ags_wave_sort_func ()

```
gint
ags_wave_sort_func (gconstpointer a,
                    gconstpointer b);
```

Compare *a* and *b* .

Parameters

a	the AgsWave	
b	another AgsWave	

Returns

0 if equal, -1 if smaller and 1 if bigger timestamp

Since: 3.0.0

ags_wave_get_audio ()

```
GObject~*
ags_wave_get_audio (AgsWave *wave);
```

Get audio.

Parameters

wave	the AgsWave	

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_wave_set_audio ()

```
void
ags_wave_set_audio (AgsWave *wave,
                    GObject *audio);
```

Set audio.

Parameters

wave	the AgsWave	
audio	the AgsAudio	

Since: 3.1.0

ags_wave_get_line ()

```
guint
ags_wave_get_line (AgsWave *wave);
```

Gets line.

Parameters

wave	the AgsWave

Returns

the line

Since: 3.1.0

ags_wave_set_line ()

```
void
ags_wave_set_line (AgsWave *wave,
                   guint line);
```

Sets line.

Parameters

wave	the AgsWave
line	the line

Since: 3.1.0

ags_wave_get_samplerate ()

```
guint
ags_wave_get_samplerate (AgsWave *wave);
```

Gets samplerate.

Parameters

wave	the AgsWave

Returns

the samplerate

Since: 3.1.0

ags_wave_set_samplerate ()

```
void
ags_wave_set_samplerate (AgsWave *wave,
                         guint samplerate);
```

Set samplerate.

Parameters

wave	the AgsWave	
samplerate	the samplerate	

Since: 3.0.0

ags_wave_get_buffer_size ()

```
guint
ags_wave_get_buffer_size (AgsWave *wave);
```

Gets buffer size.

Parameters

wave	the AgsWave	

Returns

the buffer size

Since: 3.1.0

ags_wave_set_buffer_size ()

```
void
ags_wave_set_buffer_size (AgsWave *wave,
                          guint buffer_size);
```

Set buffer size.

Parameters

wave	the AgsWave	
buffer_size	the buffer size	

Since: 3.0.0

ags_wave_get_format ()

```
guint
ags_wave_get_format (AgsWave *wave);
```

Gets format.

Parameters

wave	the AgsWave	

Returns

the format

Since: 3.1.0

ags_wave_set_format ()

```
void
ags_wave_set_format (AgsWave *wave,
                     guint format);
```

Set format.

Parameters

wave	the AgsWave	
format	the format	

Since: 3.0.0

ags_wave_get_timestamp ()

```
AgsTimestamp~*
ags_wave_get_timestamp (AgsWave *wave);
```

Get timestamp.

Parameters

wave	the AgsWave	

Returns

the AgsTimestamp.

[transfer full]

Since: 3.1.0

ags_wave_set_timestamp ()

```
void
ags_wave_set_timestamp (AgsWave *wave,
                        AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

wave	the AgsWave	
timestamp	the AgsTimestamp	

Since: 3.1.0

ags_wave_get_buffer ()

```
GList~*
ags_wave_get_buffer (AgsWave *wave);
```

Get buffer.

Parameters

wave	the AgsWave	

Returns

the GList containig AgsBuffer.

[element-type AgsAudio.Buffer][transfer full]

Since: 3.1.0

ags_wave_set_buffer ()

```
void
ags_wave_set_buffer (AgsWave *wave,
                     GList *buffer);
```

Set buffer by replacing existing.

Parameters

wave	the AgsWave	
buffer	the GList containing AgsBuffer.	*[element-type AgsAudio.Buffer][transfer full]*

Since: 3.1.0

ags_wave_add ()

```
GList~*
ags_wave_add (GList *wave,
              AgsWave *new_wave);
```

Add `new_wave` sorted to `wave`

Parameters

wave	the GList containing AgsWave.	*[element-type AgsAudio.Wave][transfer none]*
new_wave	the AgsWave to add	

Returns

the new beginning of `wave` .

[element-type AgsAudio.Wave][transfer none]

Since: 3.0.0

ags_wave_add_buffer ()

```
void
ags_wave_add_buffer (AgsWave *wave,
                     AgsBuffer *buffer,
                     gboolean use_selection_list);
```

Add *buffer* to *wave* .

Parameters

wave	the AgsWave	
buffer	the AgsBuffer to add	
use_selection_list	if TRUE add to selection, else to default wave	

Since: 3.0.0

ags_wave_remove_buffer ()

```
void
ags_wave_remove_buffer (AgsWave *wave,
                        AgsBuffer *buffer,
                        gboolean use_selection_list);
```

Removes *buffer* from *wave* .

Parameters

wave	the AgsWave	
buffer	the AgsBuffer to remove	
use_selection_list	if TRUE remove from selection, else from default wave	

Since: 3.0.0

ags_wave_get_selection ()

```
GList~*
ags_wave_get_selection (AgsWave *wave);
```

Retrieve selection.

Parameters

wave	the AgsWave	

Returns

the selection.

[element-type AgsAudio.Buffer][transfer none]

Since: 3.0.0

ags_wave_is_buffer_selected ()

```
gboolean
ags_wave_is_buffer_selected (AgsWave *wave,
                             AgsBuffer *buffer);
```

Check selection for buffer.

Parameters

wave	the AgsWave	
buffer	the AgsBuffer to check for	

Returns

TRUE if selected otherwise FALSE

Since: 3.0.0

ags_wave_find_point ()

```
AgsBuffer~*
ags_wave_find_point (AgsWave *wave,
                     guint64 x,
                     gboolean use_selection_list);
```

Find buffers by offset.

Parameters

wave	the AgsWave	
x	offset	
use_selection_list	if TRUE selection is searched	

Returns

the matching buffer as AgsBuffer.

[transfer none]

Since: 3.14.6

ags_wave_find_exact_point ()

```
AgsBuffer~*
ags_wave_find_exact_point (AgsWave *wave,
                           guint64 x,
                           gboolean use_selection_list);
```

ags_wave_find_region ()

```
GList~*
ags_wave_find_region (AgsWave *wave,
                      guint64 x0,
                      guint64 x1,
                      gboolean use_selection_list);
```

Find buffers by offset and region.

Parameters

wave	the AgsWave	
x0	x start offset	
x1	x end offset	
use_selection_list	if TRUE selection is searched	

Returns

the matching buffers as GList.

[element-type AgsAudio.Buffer][transfer container]

Since: 3.0.0

ags_wave_free_selection ()

```
void
ags_wave_free_selection (AgsWave *wave);
```

Clear selection.

Parameters

wave	the AgsWave	

Since: 3.0.0

ags_wave_free_all_selection ()

```
void
ags_wave_free_all_selection (GList *wave);
```

Clear all selection of *wave* .

Parameters

wave	the GList containing AgsWave	

Since: 3.14.10

ags_wave_add_region_to_selection ()

```
void
ags_wave_add_region_to_selection (AgsWave *wave,
                                  guint64 x0,
                                  guint64 x1,
                                  gboolean replace_current_selection);
```

Add buffer within region to selection.

Parameters

wave	the AgsWave	
x0	x start offset	
x1	x end offset	
replace_current_selection	if TRUE selection is replaced	

Since: 3.0.0

ags_wave_remove_region_from_selection ()

```
void
ags_wave_remove_region_from_selection (AgsWave *wave,
                                       guint64 x0,
                                       guint64 x1);
```

Remove buffers within region of selection.

Parameters

wave	the AgsWave	
x0	x start offset	
x1	x end offset	

Since: 3.0.0

ags_wave_add_all_to_selection ()

```
void
ags_wave_add_all_to_selection (AgsWave *wave);
```

Select all buffer to selection.

Parameters

wave	the AgsWave	

Since: 3.0.0

ags_wave_copy_selection ()

```
xmlNode~*
ags_wave_copy_selection (AgsWave *wave);
```

Copy selection to clipboard.

Parameters

wave	the AgsWave

Returns

the selection as XML.

[transfer none]

Since: 3.0.0

ags_wave_cut_selection ()

```
xmlNode~*
ags_wave_cut_selection (AgsWave *wave);
```

Cut selection to clipboard.

Parameters

wave	the AgsWave

Returns

the selection as xmlNode.

[transfer none]

Since: 3.0.0

ags_wave_insert_from_clipboard ()

```
void
ags_wave_insert_from_clipboard (AgsWave *wave,
                                xmlNode *wave_node,
                                gboolean reset_x_offset,
                                guint64 x_offset,
                                gdouble delay,
                                guint attack);
```

Paste previously copied buffers.

Parameters

wave	the AgsWave	
wave_node	the clipboard XML data	
reset_x_offset	if TRUE *x_offset* used as cursor	
x_offset	region start cursor offset	
delay	the delay to be used	
attack	the attack to be used	

Since: 3.0.0

ags_wave_insert_from_clipboard_extended ()

```
void
ags_wave_insert_from_clipboard_extended
                                (AgsWave *wave,
                                 xmlNode *wave_node,
                                 gboolean reset_x_offset,
                                 guint64 x_offset,
                                 gdouble delay,
                                 guint attack,
                                 gboolean match_line,
                                 gboolean do_replace);
```

Paste previously copied buffers.

Parameters

wave	the AgsWave	
wave_node	the clipboard XML data	
reset_x_offset	if TRUE *x_offset* used as cursor	
x_offset	region start cursor offset	
delay	the delay to be used	
attack	the attack to be used	
match_line	only paste if channel matches	
do_replace	if TRUE current data is replaced, otherwise additive mixing is performed	

Since: 3.0.0

ags_wave_new ()

```
AgsWave~*
ags_wave_new (GObject *audio,
              guint line);
```

Creates a new instance of AgsWave.

Parameters

audio	the assigned AgsAudio	
line	the line to be used	

Returns

a new AgsWave

Since: 3.0.0

AGS_IS_WAVE()

```
#define AGS_IS_WAVE(obj)                (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_WAVE))
```

AGS_IS_WAVE_CLASS()

```
#define AGS_IS_WAVE_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_WAVE))
```

AGS_WAVE()

```
#define AGS_WAVE(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_WAVE,  ↩
    AgsWave))
```

AGS_WAVE_CLASS()

```
#define AGS_WAVE_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_WAVE,  ↩
    AgsWaveClass))
```

AGS_WAVE_GET_CLASS()

```
#define AGS_WAVE_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS (obj, AGS_TYPE_WAVE,  ↩
    AgsWaveClass))
```

ags_wave_flags_get_type ()

```
GType
ags_wave_flags_get_type ();
```

ags_wave_get_type ()

```
GType
ags_wave_get_type (void);
```

Types and Values

AGS_WAVE_DEFAULT_BPM

```
#define AGS_WAVE_DEFAULT_BPM (120.0)
```

AGS_WAVE_TICS_PER_BEAT

```
#define AGS_WAVE_TICS_PER_BEAT (1.0)
```

AGS_WAVE_MINIMUM_BUFFER_LENGTH

```
#define AGS_WAVE_MINIMUM_BUFFER_LENGTH (1.0 / 16.0)
```

AGS_WAVE_DEFAULT_BUFFER_LENGTH

```
#define AGS_WAVE_DEFAULT_BUFFER_LENGTH (64.0)
```

AGS_WAVE_DEFAULT_LENGTH

```
#define AGS_WAVE_DEFAULT_LENGTH ((60 * AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_WAVE_DEFAULT_BPM / 4) * 16 * 16 * 1200 / AGS_WAVE_TICS_PER_BEAT)
```

AGS_WAVE_DEFAULT_JIFFIE

```
#define AGS_WAVE_DEFAULT_JIFFIE (60.0 / AGS_WAVE_DEFAULT_BPM / AGS_WAVE_TICS_PER_BEAT)
```

AGS_WAVE_DEFAULT_DURATION

```
#define AGS_WAVE_DEFAULT_DURATION (AGS_WAVE_DEFAULT_LENGTH * AGS_WAVE_DEFAULT_JIFFIE *  ↩
    AGS_USEC_PER_SEC)
```

AGS_WAVE_DEFAULT_OFFSET

```
#define AGS_WAVE_DEFAULT_OFFSET (AGS_WAVE_DEFAULT_BUFFER_LENGTH *  ↩
    AGS_SOUNDCARD_DEFAULT_SAMPLERATE)
```

AGS_WAVE_CLIPBOARD_VERSION

```
#define AGS_WAVE_CLIPBOARD_VERSION "1.4.0"
```

AGS_WAVE_CLIPBOARD_TYPE

```
#define AGS_WAVE_CLIPBOARD_TYPE "AgsWaveClipboardXml"
```

AGS_WAVE_CLIPBOARD_FORMAT

```
#define AGS_WAVE_CLIPBOARD_FORMAT "AgsWaveNativeLevel"
```

enum AgsWaveFlags

Enum values to control the behavior or indicate internal state of AgsWave by enable/disable as flags.

Members

AGS_WAVE_BYPASS	ignore any wave data

AGS_TYPE_WAVE

```
#define AGS_TYPE_WAVE                (ags_wave_get_type())
```

struct AgsWave

```
struct AgsWave;
```

struct AgsWaveClass

```
struct AgsWaveClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsWave

Flags: Read / Write

Since: 3.0.0

The "buffer" property

```
  "buffer"                   gpointer
```

The assigned AgsBuffer

[transfer full]

Owner: AgsWave

Flags: Read / Write

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The audio buffer's buffer size.

Owner: AgsWave

Flags: Read / Write

Default value: 1024

Since: 3.0.0

The "format" property

```
"format"                     guint
```

The audio buffer's format.

Owner: AgsWave

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "line" property

```
"line"                       guint
```

The wave's line.

Owner: AgsWave

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "samplerate" property

```
"samplerate"                 guint
```

The audio buffer's samplerate.

Owner: AgsWave

Flags: Read / Write

Default value: 48000

Since: 3.0.0

The "timestamp" property

```
"timestamp"                  AgsTimestamp~*
```

The pattern's timestamp.

Owner: AgsWave

Flags: Read / Write

Since: 3.0.0

10.6 AgsBuffer

AgsBuffer — Buffer class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_BUFFER_GET_OBJ_MUTEX()
GRecMutex *	ags_buffer_get_obj_mutex ()
void	ags_buffer_lock ()
void	ags_buffer_unlock ()
gboolean	ags_buffer_test_flags ()
void	ags_buffer_set_flags ()
void	ags_buffer_unset_flags ()
gint	ags_buffer_sort_func ()
guint64	ags_buffer_get_x ()
void	ags_buffer_set_x ()
guint	ags_buffer_get_samplerate ()
void	ags_buffer_set_samplerate ()
guint	ags_buffer_get_buffer_size ()
void	ags_buffer_set_buffer_size ()
guint	ags_buffer_get_format ()
void	ags_buffer_set_format ()
gpointer	ags_buffer_get_data ()
AgsBuffer *	ags_buffer_duplicate ()
AgsBuffer *	ags_buffer_new ()
#define	AGS_BUFFER()
#define	AGS_BUFFER_CLASS()
#define	AGS_BUFFER_GET_CLASS()
#define	AGS_IS_BUFFER()
#define	AGS_IS_BUFFER_CLASS()
GType	ags_buffer_flags_get_type ()
GType	ags_buffer_get_type ()

Properties

guint	buffer-size	Read / Write
gpointer	data	Read / Write
guint	format	Read / Write
guint	samplerate	Read / Write
guint64	selection-x0	Read / Write
guint64	selection-x1	Read / Write
guint64	x	Read / Write

Types and Values

enum	AgsBufferFlags
#define	AGS_TYPE_BUFFER

struct	AgsBuffer
struct	AgsBufferClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsBufferFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsBuffer
```

Includes

```
#include <ags/audio/ags_buffer.h>
```

Description

AgsBuffer represents audio data. If you want to modify `data` property you have to lock the buffer using ags_buffer_lock() and when you are done release it using ags_buffer_unlock().

The `x` property is the offset in frames matching samplerate.

Functions

AGS_BUFFER_GET_OBJ_MUTEX()

```
#define AGS_BUFFER_GET_OBJ_MUTEX(obj) (&(((AgsBuffer *) obj)->obj_mutex))
```

ags_buffer_get_obj_mutex ()

```
GRecMutex~*
ags_buffer_get_obj_mutex (AgsBuffer *buffer);
```

Get object mutex.

Parameters

buffer	the AgsBuffer

Returns

the GRecMutex to lock *buffer*

Since: 3.1.0

ags_buffer_lock ()

```
void
ags_buffer_lock (AgsBuffer *buffer);
```

Lock object mutex.

Parameters

buffer	the AgsBuffer

Since: 3.1.0

ags_buffer_unlock ()

```
void
ags_buffer_unlock (AgsBuffer *buffer);
```

Unlock object mutex.

Parameters

buffer	the AgsBuffer

Since: 3.1.0

ags_buffer_test_flags ()

```
gboolean
ags_buffer_test_flags (AgsBuffer *buffer,
                       guint flags);
```

Test *flags* to be set on *buffer* .

Parameters

buffer	the AgsBuffer
flags	the flags

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_buffer_set_flags ()

```
void
ags_buffer_set_flags (AgsBuffer *buffer,
                      guint flags);
```

Set *flags* on *buffer* .

Parameters

buffer	the AgsBuffer
flags	the flags

Since: 3.0.0

ags_buffer_unset_flags ()

```
void
ags_buffer_unset_flags (AgsBuffer *buffer,
                        guint flags);
```

Unset *flags* on *buffer* .

Parameters

buffer	the AgsBuffer	
flags	the flags	

Since: 3.0.0

ags_buffer_sort_func ()

```
gint
ags_buffer_sort_func (gconstpointer a,
                      gconstpointer b);
```

Sort buffers.

Parameters

a	an AgsBuffer	
b	an other AgsBuffer	

Returns

0 if equal, -1 if smaller and 1 if bigger offset

Since: 3.0.0

ags_buffer_get_x ()

```
guint64
ags_buffer_get_x (AgsBuffer *buffer);
```

Gets x.

Parameters

buffer	the AgsBuffer	

Returns

the x

Since: 3.1.0

ags_buffer_set_x ()

```
void
ags_buffer_set_x (AgsBuffer *buffer,
                  guint64 x);
```

Sets x.

Parameters

buffer	the AgsBuffer	
x	the x	

Since: 3.1.0

ags_buffer_get_samplerate ()

```
guint
ags_buffer_get_samplerate (AgsBuffer *buffer);
```

Gets samplerate.

Parameters

buffer	the AgsBuffer	

Returns

the samplerate

Since: 3.1.0

ags_buffer_set_samplerate ()

```
void
ags_buffer_set_samplerate (AgsBuffer *buffer,
                           guint samplerate);
```

Set samplerate.

Parameters

buffer	the AgsBuffer	
samplerate	the samplerate	

Since: 3.0.0

ags_buffer_get_buffer_size ()

```
guint
ags_buffer_get_buffer_size (AgsBuffer *buffer);
```

Gets buffer size.

Parameters

buffer	the AgsBuffer	

Returns

the buffer size

Since: 3.1.0

ags_buffer_set_buffer_size ()

```
void
ags_buffer_set_buffer_size (AgsBuffer *buffer,
                            guint buffer_size);
```

Set buffer size.

Parameters

buffer	the AgsBuffer	
buffer_size	the buffer size	

Since: 3.0.0

ags_buffer_get_format ()

```
guint
ags_buffer_get_format (AgsBuffer *buffer);
```

Gets format.

Parameters

buffer	the AgsBuffer	

Returns

the format

Since: 3.1.0

ags_buffer_set_format ()

```
void
ags_buffer_set_format (AgsBuffer *buffer,
                       guint format);
```

Set format.

Parameters

buffer	the AgsBuffer	
format	the format	

Since: 3.0.0

ags_buffer_get_data ()

```
gpointer
ags_buffer_get_data (AgsBuffer *buffer);
```

Gets data.

Parameters

buffer	the AgsBuffer	

Returns

the data

Since: 3.1.0

ags_buffer_duplicate ()

```
AgsBuffer~*
ags_buffer_duplicate (AgsBuffer *buffer);
```

Duplicate a buffer.

Parameters

buffer	an AgsBuffer	

Returns

the duplicated AgsBuffer.

[transfer full]

Since: 3.0.0

ags_buffer_new ()

```
AgsBuffer~*
ags_buffer_new ();
```

Creates a new instance of AgsBuffer

Returns

the new AgsBuffer

Since: 3.0.0

AGS_BUFFER()

```
#define AGS_BUFFER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_BUFFER, ↩
    AgsBuffer))
```

AGS_BUFFER_CLASS()

```
#define AGS_BUFFER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_BUFFER, ↩
    AgsBufferClass))
```

AGS_BUFFER_GET_CLASS()

```
#define AGS_BUFFER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_BUFFER, ↩
    AgsBufferClass))
```

AGS_IS_BUFFER()

```
#define AGS_IS_BUFFER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_BUFFER))
```

AGS_IS_BUFFER_CLASS()

```
#define AGS_IS_BUFFER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_BUFFER))
```

ags_buffer_flags_get_type ()

```
GType
ags_buffer_flags_get_type ();
```

ags_buffer_get_type ()

```
GType
ags_buffer_get_type ();
```

Types and Values

enum AgsBufferFlags

Enum values to control the behavior or indicate internal state of AgsBuffer by enable/disable as flags.

Members

AGS_BUFFER_IS_SELECTED	is selected

AGS_TYPE_BUFFER

```
#define AGS_TYPE_BUFFER                 (ags_buffer_get_type())
```

struct AgsBuffer

```
struct AgsBuffer;
```

struct AgsBufferClass

```
struct AgsBufferClass {
  GObjectClass gobject;
};
```

Property Details

The "buffer-size" property

```
  "buffer-size"               guint
```

Buffer's audio data buffer size.

Owner: AgsBuffer

Flags: Read / Write

Default value: 1024

Since: 3.0.0

The "data" property

```
  "data"                      gpointer
```

Buffer's audio data.

Owner: AgsBuffer

Flags: Read / Write

Since: 3.0.0

The "format" property

```
  "format"                    guint
```

Buffer's audio data format.

Owner: AgsBuffer

Flags: Read / Write

Default value: 16

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

Buffer's audio data samplerate.

Owner: AgsBuffer

Flags: Read / Write

Default value: 48000

Since: 3.0.0

The "selection-x0" property

```
"selection-x0"              guint64
```

Buffer's selection x0 offset.

Owner: AgsBuffer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "selection-x1" property

```
"selection-x1"              guint64
```

Buffer's selection x1 offset.

Owner: AgsBuffer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "x" property

```
"x"                         guint64
```

Buffer's x offset.

Owner: AgsBuffer

Flags: Read / Write

Default value: 0

Since: 3.0.0

10.7 AgsMidi

AgsMidi — Midi class supporting selection and clipboard.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MIDI_GET_OBJ_MUTEX()
GRecMutex *	ags_midi_get_obj_mutex ()
gboolean	ags_midi_test_flags ()
void	ags_midi_set_flags ()
void	ags_midi_unset_flags ()
GList *	ags_midi_find_near_timestamp ()
gint	ags_midi_sort_func ()
GObject *	ags_midi_get_audio ()
void	ags_midi_set_audio ()
guint	ags_midi_get_audio_channel ()
void	ags_midi_set_audio_channel ()
AgsTimestamp *	ags_midi_get_timestamp ()
void	ags_midi_set_timestamp ()
GList *	ags_midi_get_track ()
void	ags_midi_set_track ()
GList *	ags_midi_add ()
void	ags_midi_add_track ()
void	ags_midi_remove_track ()
AgsMidi *	ags_midi_new ()
#define	AGS_IS_MIDI()
#define	AGS_IS_MIDI_CLASS()
#define	AGS_MIDI()
#define	AGS_MIDI_CLASS()
#define	AGS_MIDI_GET_CLASS()
GType	ags_midi_get_type ()

Properties

AgsAudio *	audio	Read / Write
guint	audio-channel	Read / Write
AgsTimestamp *	timestamp	Read / Write
gpointer	track	Read / Write

Types and Values

#define	AGS_MIDI_DEFAULT_BPM
#define	AGS_MIDI_TICS_PER_BEAT
#define	AGS_MIDI_MINIMUM_TRACK_LENGTH
#define	AGS_MIDI_DEFAULT_LENGTH
#define	AGS_MIDI_DEFAULT_JIFFIE
#define	AGS_MIDI_DEFAULT_DURATION
#define	AGS_MIDI_DEFAULT_OFFSET
#define	AGS_MIDI_CLIPBOARD_VERSION
#define	AGS_MIDI_CLIPBOARD_TYPE
#define	AGS_MIDI_CLIPBOARD_FORMAT
enum	AgsMidiFlags

#define	AGS_TYPE_MIDI
struct	AgsMidi
struct	AgsMidiClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsMidi
```

Includes

```
#include <ags/audio/ags_midi.h>
```

Description

AgsMidi acts as a container of AgsTrack.

Functions

AGS_MIDI_GET_OBJ_MUTEX()

```
#define AGS_MIDI_GET_OBJ_MUTEX(obj) (&(((AgsMidi *) obj)->obj_mutex))
```

ags_midi_get_obj_mutex ()

```
GRecMutex~*
ags_midi_get_obj_mutex (AgsMidi *midi);
```

Get object mutex.

Parameters

midi	the AgsMidi

Returns

the GRecMutex to lock *midi*

Since: 3.1.0

ags_midi_test_flags ()

```
gboolean
ags_midi_test_flags (AgsMidi *midi,
                     guint flags);
```

Test *flags* to be set on *midi* .

Parameters

midi	the AgsMidi	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_midi_set_flags ()

```
void
ags_midi_set_flags (AgsMidi *midi,
                    guint flags);
```

Set *flags* on *midi* .

Parameters

midi	the AgsMidi	
flags	the flags	

Since: 3.0.0

ags_midi_unset_flags ()

```
void
ags_midi_unset_flags (AgsMidi *midi,
                      guint flags);
```

Unset *flags* on *midi* .

Parameters

midi	the AgsMidi	
flags	the flags	

Since: 3.0.0

ags_midi_find_near_timestamp ()

```
GList~*
ags_midi_find_near_timestamp (GList *midi,
                              guint audio_channel,
                              AgsTimestamp *timestamp);
```

Retrieve appropriate midi for timestamp.

Parameters

midi	the GList containing AgsMidi.	*[element-type AgsAudio.Midi][transfer none]*
audio_channel	the matching audio channel	
timestamp	the matching timestamp, or NULL to match any timestamp.	*[allow-none]*

Returns

Next match.

[element-type AgsAudio.Midi][transfer none]

Since: 3.0.0

ags_midi_sort_func ()

```
gint
ags_midi_sort_func (gconstpointer a,
                    gconstpointer b);
```

Compare `a` and `b` .

Parameters

a	the AgsMidi	
b	another AgsMidi	

Returns

0 if equal, -1 if smaller and 1 if bigger timestamp

Since: 3.0.0

ags_midi_get_audio ()

```
GObject~*
ags_midi_get_audio (AgsMidi *midi);
```

Get audio.

Parameters

midi	the AgsMidi	

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_midi_set_audio ()

```
void
ags_midi_set_audio (AgsMidi *midi,
                    GObject *audio);
```

Set audio.

Parameters

midi	the AgsMidi	
audio	the AgsAudio	

Since: 3.1.0

ags_midi_get_audio_channel ()

```
guint
ags_midi_get_audio_channel (AgsMidi *midi);
```

Gets audio channel.

Parameters

midi	the AgsMidi	

Returns

the audio channel

Since: 3.1.0

ags_midi_set_audio_channel ()

```
void
ags_midi_set_audio_channel (AgsMidi *midi,
                            guint audio_channel);
```

Sets audio channel.

Parameters

midi	the AgsMidi	
audio_channel	the audio channel	

Since: 3.1.0

ags_midi_get_timestamp ()

```
AgsTimestamp~*
ags_midi_get_timestamp (AgsMidi *midi);
```

Get timestamp.

Parameters

midi	the AgsMidi

Returns

the AgsTimestamp.

[transfer full]

Since: 3.1.0

ags_midi_set_timestamp ()

```
void
ags_midi_set_timestamp (AgsMidi *midi,
                        AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

midi	the AgsMidi
timestamp	the AgsTimestamp

Since: 3.1.0

ags_midi_get_track ()

```
GList~*
ags_midi_get_track (AgsMidi *midi);
```

Get track.

Parameters

midi	the AgsMidi

Returns

the GList containig AgsTrack.

[element-type AgsAudio.Track][transfer full]

Since: 3.1.0

ags_midi_set_track ()

```
void
ags_midi_set_track (AgsMidi *midi,
                    GList *track);
```

Set track by replacing existing.

Parameters

midi	the AgsMidi	
track	the GList containing AgsTrack.	*[element-type AgsAudio.Track][transfer full]*

Since: 3.1.0

ags_midi_add ()

```
GList~*
ags_midi_add (GList *midi,
              AgsMidi *new_midi);
```

Add `new_midi` sorted to `midi`

Parameters

midi	the GList containing AgsMidi.	*[element-type AgsAudio.Midi][transfer none]*
new_midi	the midi to add	

Returns

the new beginning of `midi` .

[element-type AgsAudio.Midi][transfer none]

Since: 3.0.0

ags_midi_add_track ()

```
void
ags_midi_add_track (AgsMidi *midi,
                    AgsTrack *track,
                    gboolean use_selection_list);
```

Adds a track to midi.

Parameters

midi	an AgsMidi	
track	the AgsTrack to add	
use_selection_list	if TRUE add to selection, else to default midi	

Since: 3.0.0

ags_midi_remove_track ()

```
void
ags_midi_remove_track (AgsMidi *midi,
                       AgsTrack *track,
                       gboolean use_selection_list);
```

Removes a track from midi.

Parameters

midi	an AgsMidi	
track	the AgsTrack to remove	
use_selection_list	if TRUE remove from selection, else from default midi	

Since: 3.0.0

ags_midi_new ()

```
AgsMidi~*
ags_midi_new (GObject *audio,
              guint audio_channel);
```

Creates a new instance of AgsMidi.

Parameters

audio	the assigned AgsAudio	
audio_channel	the audio channel to be used	

Returns

the new AgsMidi

Since: 3.0.0

AGS_IS_MIDI()

```
#define AGS_IS_MIDI(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_MIDI))
```

AGS_IS_MIDI_CLASS()

```
#define AGS_IS_MIDI_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_MIDI))
```

AGS_MIDI()

```
#define AGS_MIDI(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_MIDI,  ↩
    AgsMidi))
```

AGS_MIDI_CLASS()

```
#define AGS_MIDI_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_MIDI,  ↩
    AgsMidiClass))
```

AGS_MIDI_GET_CLASS()

```
#define AGS_MIDI_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (obj, AGS_TYPE_MIDI,  ↩
    AgsMidiClass))
```

ags_midi_get_type ()

```
GType
ags_midi_get_type (void);
```

Types and Values

AGS_MIDI_DEFAULT_BPM

```
#define AGS_MIDI_DEFAULT_BPM (120.0)
```

AGS_MIDI_TICS_PER_BEAT

```
#define AGS_MIDI_TICS_PER_BEAT (1.0)
```

AGS_MIDI_MINIMUM_TRACK_LENGTH

```
#define AGS_MIDI_MINIMUM_TRACK_LENGTH (1.0 / 16.0)
```

AGS_MIDI_DEFAULT_LENGTH

```
#define AGS_MIDI_DEFAULT_LENGTH (65535.0 / AGS_MIDI_TICS_PER_BEAT -  ↩
    AGS_MIDI_MINIMUM_TRACK_LENGTH)
```

AGS_MIDI_DEFAULT_JIFFIE

```
#define AGS_MIDI_DEFAULT_JIFFIE (60.0 / AGS_MIDI_DEFAULT_BPM / AGS_MIDI_TICS_PER_BEAT)
```

AGS_MIDI_DEFAULT_DURATION

```
#define AGS_MIDI_DEFAULT_DURATION (AGS_MIDI_DEFAULT_LENGTH * AGS_MIDI_DEFAULT_JIFFIE *  ↩
    AGS_USEC_PER_SEC)
```

AGS_MIDI_DEFAULT_OFFSET

```
#define AGS_MIDI_DEFAULT_OFFSET (64 * (1 / AGS_MIDI_MINIMUM_TRACK_LENGTH))
```

AGS_MIDI_CLIPBOARD_VERSION

```
#define AGS_MIDI_CLIPBOARD_VERSION "1.4.0"
```

AGS_MIDI_CLIPBOARD_TYPE

```
#define AGS_MIDI_CLIPBOARD_TYPE "AgsMidiClipboardXml"
```

AGS_MIDI_CLIPBOARD_FORMAT

```
#define AGS_MIDI_CLIPBOARD_FORMAT "AgsMidiNativeLevel"
```

enum AgsMidiFlags

Enum values to control the behavior or indicate internal state of AgsMidi by enable/disable as flags.

Members

AGS_MIDI_BYPASS	ignore any midi data

AGS_TYPE_MIDI

```
#define AGS_TYPE_MIDI                (ags_midi_get_type())
```

struct AgsMidi

```
struct AgsMidi;
```

struct AgsMidiClass

```
struct AgsMidiClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsMidi

Flags: Read / Write

Since: 3.0.0

The "audio-channel" property

```
"audio-channel"            guint
```

The midi's audio-channel.

Owner: AgsMidi

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "timestamp" property

```
"timestamp"                AgsTimestamp~*
```

The midi's timestamp.

Owner: AgsMidi

Flags: Read / Write

Since: 3.0.0

The "track" property

```
"track"                    gpointer
```

The assigned AgsTrack

[transfer full]

Owner: AgsMidi

Flags: Read / Write

Since: 3.0.0

10.8 AgsTrack

AgsTrack — Track class.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TRACK_GET_OBJ_MUTEX()
GRecMutex *	ags_track_get_obj_mutex ()
void	ags_track_lock ()
void	ags_track_unlock ()
gboolean	ags_track_test_flags ()
void	ags_track_set_flags ()

void	ags_track_unset_flags ()
gint	ags_track_sort_func ()
guint64	ags_track_get_x ()
void	ags_track_set_x ()
gpointer	ags_track_get_smf_buffer ()
gpointer	ags_track_alloc_smf_buffer ()
gpointer	ags_track_realloc_smf_buffer ()
AgsTrack *	ags_track_duplicate ()
AgsTrack *	ags_track_new ()
#define	AGS_IS_TRACK()
#define	AGS_IS_TRACK_CLASS()
#define	AGS_TRACK()
#define	AGS_TRACK_CLASS()
#define	AGS_TRACK_GET_CLASS()
GType	ags_track_flags_get_type ()
GType	ags_track_get_type ()

Properties

gpointer	smf-buffer	Read / Write
guint64	x	Read / Write

Types and Values

enum	AgsTrackFlags
#define	AGS_TYPE_TRACK
struct	AgsTrack
struct	AgsTrackClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsTrackFlags
GObject
&#x2570;&#x2500;&#x2500; AgsTrack
```

Includes

```
#include <ags/audio/ags_track.h>
```

Description

AgsTrack represents a tone.

Functions

AGS_TRACK_GET_OBJ_MUTEX()

```
#define AGS_TRACK_GET_OBJ_MUTEX(obj) (&(((AgsTrack *) obj)->obj_mutex))
```

ags_track_get_obj_mutex ()

```
GRecMutex~*
ags_track_get_obj_mutex (AgsTrack *track);
```

Get object mutex.

Parameters

track	the AgsTrack

Returns

the GRecMutex to lock *track*

Since: 3.1.0

ags_track_lock ()

```
void
ags_track_lock (AgsTrack *track);
```

Lock object mutex.

Parameters

track	the AgsTrack

Since: 3.1.0

ags_track_unlock ()

```
void
ags_track_unlock (AgsTrack *track);
```

Unlock object mutex.

Parameters

track	the AgsTrack

Since: 3.1.0

ags_track_test_flags ()

```
gboolean
ags_track_test_flags (AgsTrack *track,
                      guint flags);
```

Test *flags* to be set on *track* .

Parameters

track	the AgsTrack	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_track_set_flags ()

```
void
ags_track_set_flags (AgsTrack *track,
                     guint flags);
```

Set *flags* on *track* .

Parameters

track	the AgsTrack	
flags	the flags	

Since: 3.0.0

ags_track_unset_flags ()

```
void
ags_track_unset_flags (AgsTrack *track,
                       guint flags);
```

Unset *flags* on *track* .

Parameters

track	the AgsTrack	
flags	the flags	

Since: 3.0.0

ags_track_sort_func ()

```
gint
ags_track_sort_func (gconstpointer a,
                     gconstpointer b);
```

Sort tracks.

Parameters

a	an AgsTrack	
b	an AgsTrack	

Returns

0 if equal, -1 if smaller and 1 if bigger offset

Since: 3.0.0

ags_track_get_x ()

```
guint64
ags_track_get_x (AgsTrack *track);
```

Gets x.

Parameters

track	the AgsTrack	

Returns

the x

Since: 3.1.0

ags_track_set_x ()

```
void
ags_track_set_x (AgsTrack *track,
                 guint64 x);
```

Sets x.

Parameters

track	the AgsTrack	
x	the x	

Since: 3.1.0

ags_track_get_smf_buffer ()

```
gpointer
ags_track_get_smf_buffer (AgsTrack *track,
                          guint *smf_buffer_length);
```

ags_track_alloc_smf_buffer ()

```
gpointer
ags_track_alloc_smf_buffer (AgsTrack *track,
                            guint smf_buffer_length);
```

Allocate SMF buffer of *track* .

Parameters

track	the AgsTrack	
smf_buffer_length	SMF buffer length	

Returns

the newly allocated SMF buffer

Since: 3.6.17

ags_track_realloc_smf_buffer ()

```
gpointer
ags_track_realloc_smf_buffer (AgsTrack *track,
                              guint smf_buffer_length);
```

Reallocate SMF buffer of *track* .

Parameters

track	the AgsTrack	
smf_buffer_length	SMF buffer length	

Returns

the reallocated SMF buffer

Since: 3.6.17

ags_track_duplicate ()

```
AgsTrack~*
ags_track_duplicate (AgsTrack *track);
```

Duplicate a track.

Parameters

track	an AgsTrack	

Returns

the duplicated AgsTrack.

[transfer full]

Since: 3.0.0

ags_track_new ()

```
AgsTrack~*
ags_track_new ();
```

Creates a new instance of AgsTrack.

Returns

the new AgsTrack

Since: 3.0.0

AGS_IS_TRACK()

```
#define AGS_IS_TRACK(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_TRACK))
```

AGS_IS_TRACK_CLASS()

```
#define AGS_IS_TRACK_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_TRACK))
```

AGS_TRACK()

```
#define AGS_TRACK(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_TRACK,  ↩
    AgsTrack))
```

AGS_TRACK_CLASS()

```
#define AGS_TRACK_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_TRACK,  ↩
    AgsTrackClass))
```

AGS_TRACK_GET_CLASS()

```
#define AGS_TRACK_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_TRACK,  ↩
    AgsTrackClass))
```

ags_track_flags_get_type ()

```
GType
ags_track_flags_get_type ();
```

ags_track_get_type ()

```
GType
ags_track_get_type ();
```

Types and Values

enum AgsTrackFlags

Enum values to control the behavior or indicate internal state of AgsTrack by enable/disable as flags.

Members

AGS_TRACK_IS_SELECTED	is selected

AGS_TYPE_TRACK

```
#define AGS_TYPE_TRACK                (ags_track_get_type())
```

struct AgsTrack

```
struct AgsTrack;
```

struct AgsTrackClass

```
struct AgsTrackClass {
  GObjectClass gobject;
};
```

Property Details

The "smf-buffer" property

```
  "smf-buffer"               gpointer
```

Track's SMF buffer.

Owner: AgsTrack

Flags: Read / Write

Since: 3.0.0

The "x" property

```
"x"                         guint64
```

Track's x offset.

Owner: AgsTrack

Flags: Read / Write

Default value: 0

Since: 3.0.0

10.9 AgsPattern

AgsPattern — Pattern representing tones

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PATTERN_GET_OBJ_MUTEX()
GRecMutex *	ags_pattern_get_obj_mutex ()
gboolean	ags_pattern_test_flags ()
void	ags_pattern_set_flags ()
void	ags_pattern_unset_flags ()
GList *	ags_pattern_find_near_timestamp ()
GObject *	ags_pattern_get_channel ()
void	ags_pattern_set_channel ()
AgsTimestamp *	ags_pattern_get_timestamp ()
void	ags_pattern_set_timestamp ()
void	ags_pattern_get_dim ()
void	ags_pattern_set_dim ()
GObject *	ags_pattern_get_port ()
void	ags_pattern_set_port ()
gboolean	ags_pattern_is_empty ()
gboolean	ags_pattern_get_bit ()
void	ags_pattern_toggle_bit ()
AgsNote *	ags_pattern_get_note ()
AgsPattern *	ags_pattern_new ()
#define	AGS_IS_PATTERN()
#define	AGS_IS_PATTERN_CLASS()
#define	AGS_PATTERN()
#define	AGS_PATTERN_CLASS()
#define	AGS_PATTERN_GET_CLASS()
GType	ags_pattern_get_type ()

Properties

AgsChannel *	channel	Read / Write

gboolean	current-bit	Read
guint	first-index	Read / Write
guint	offset	Read / Write
AgsPort *	port	Read / Write
guint	second-index	Read / Write
AgsTimestamp *	timestamp	Read / Write

Types and Values

#define	AGS_PATTERN_DEFAULT_BPM
#define	AGS_PATTERN_TICS_PER_BEAT
#define	AGS_PATTERN_MINIMUM_NOTE_LENGTH
#define	AGS_PATTERN_MAXIMUM_NOTE_LENGTH
#define	AGS_PATTERN_DEFAULT_LENGTH
#define	AGS_PATTERN_DEFAULT_JIFFIE
#define	AGS_PATTERN_DEFAULT_DURATION
#define	AGS_PATTERN_DEFAULT_OFFSET
#define	AGS_TYPE_PATTERN
struct	AgsPattern
struct	AgsPatternClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsPattern
```

Implemented Interfaces

AgsPattern implements AgsConnectable and AgsTactable.

Includes

```
#include <ags/audio/ags_pattern.h>
```

Description

AgsPattern represents an audio pattern of tones. The bitmask pattern is stored in has 2 level of banks.

You can check if a bit is set by calling ags_pattern_get_bit().

To enable/disable a bit call ags_pattern_toggle_bit().

Functions

AGS_PATTERN_GET_OBJ_MUTEX()

```
#define AGS_PATTERN_GET_OBJ_MUTEX(obj) (&(((AgsPattern *) obj)->obj_mutex))
```

ags_pattern_get_obj_mutex ()

```
GRecMutex~*
ags_pattern_get_obj_mutex (AgsPattern *pattern);
```

Get object mutex.

Parameters

pattern	the AgsPattern	

Returns

the GRecMutex to lock *pattern*

Since: 3.1.0

ags_pattern_test_flags ()

```
gboolean
ags_pattern_test_flags (AgsPattern *pattern,
                        guint flags);
```

Test *flags* to be set on *pattern* .

Parameters

pattern	the AgsPattern	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_pattern_set_flags ()

```
void
ags_pattern_set_flags (AgsPattern *pattern,
                       guint flags);
```

Set flags.

Parameters

pattern	the AgsPattern	
flags	the flags	

Since: 3.0.0

ags_pattern_unset_flags ()

```
void
ags_pattern_unset_flags (AgsPattern *pattern,
                         guint flags);
```

Unset flags.

Parameters

pattern	the AgsPattern	
flags	the flags	

Since: 3.0.0

ags_pattern_find_near_timestamp ()

```
GList~*
ags_pattern_find_near_timestamp (GList *pattern,
                                 AgsTimestamp *timestamp);
```

Retrieve appropriate pattern for timestamp.

Parameters

pattern	the GList containing AgsPattern.	*[element-type AgsAudio.Pattern][transfer none]*
timestamp	the matching AgsTimestamp	

Returns

Next match.

[element-type AgsAudio.Pattern][transfer none]

Since: 3.0.0

ags_pattern_get_channel ()

```
GObject~*
ags_pattern_get_channel (AgsPattern *pattern);
```

Get channel.

Parameters

pattern	the AgsPattern	

Returns

the AgsChannel.

[transfer full]

Since: 3.1.0

ags_pattern_set_channel ()

```
void
ags_pattern_set_channel (AgsPattern *pattern,
                         GObject *channel);
```

Set channel.

Parameters

pattern	the AgsPattern	
channel	the AgsChannel	

Since: 3.1.0

ags_pattern_get_timestamp ()

```
AgsTimestamp~*
ags_pattern_get_timestamp (AgsPattern *pattern);
```

Get timestamp.

Parameters

pattern	the AgsPattern	

Returns

the AgsTimestamp.

[transfer full]

Since: 3.1.0

ags_pattern_set_timestamp ()

```
void
ags_pattern_set_timestamp (AgsPattern *pattern,
                           AgsTimestamp *timestamp);
```

Set timestamp.

Parameters

pattern	the AgsPattern	
timestamp	the AgsTimestamp	

Since: 3.1.0

ags_pattern_get_dim ()

```
void
ags_pattern_get_dim (AgsPattern *pattern,
                     guint *dim0,
                     guint *dim1,
                     guint *length);
```

Get the pattern's dimensions.

Parameters

pattern	an AgsPattern	
dim0	bank 0 size.	*[out]*
dim1	bank 1 size.	*[out]*
length	amount of beats.	*[out]*

Since: 3.1.0

ags_pattern_set_dim ()

```
void
ags_pattern_set_dim (AgsPattern *pattern,
                     guint dim0,
                     guint dim1,
                     guint length);
```

Reallocates the pattern's dimensions.

Parameters

pattern	an AgsPattern	
dim0	bank 0 size	
dim1	bank 1 size	
length	amount of beats	

Since: 3.0.0

ags_pattern_get_port ()

```
GObject~*
ags_pattern_get_port (AgsPattern *pattern);
```

ags_pattern_set_port ()

```
void
ags_pattern_set_port (AgsPattern *pattern,
                      GObject *port);
```

ags_pattern_is_empty ()

```
gboolean
ags_pattern_is_empty (AgsPattern *pattern,
                      guint i,
                      guint j);
```

ags_pattern_get_bit ()

```
gboolean
ags_pattern_get_bit (AgsPattern *pattern,
                     guint i,
                     guint j,
                     guint bit);
```

Check for tic to be played.

Parameters

pattern	an AgsPattern	
i	bank index 0	
j	bank index 1	
bit	the tic to check	

Returns

TRUE if tone is enabled.

Since: 3.0.0

ags_pattern_toggle_bit ()

```
void
ags_pattern_toggle_bit (AgsPattern *pattern,
                        guint i,
                        guint j,
                        guint bit);
```

Toggle tone.

Parameters

pattern	an AgsPattern	
i	bank index 0	
j	bank index 1	
bit	the tic to toggle	

Since: 3.0.0

ags_pattern_get_note ()

```
AgsNote~*
ags_pattern_get_note (AgsPattern *pattern,
                      guint bit);
```

Get note of *pattern* .

Parameters

pattern	an AgsPattern	
bit	the tic	

Returns

the AgsNote at given position or NULL.

[transfer full]

Since: 3.3.0

ags_pattern_new ()

```
AgsPattern~*
ags_pattern_new ();
```

Creates a new instance of AgsPattern

Returns

the new AgsPattern

Since: 3.0.0

AGS_IS_PATTERN()

```
#define AGS_IS_PATTERN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PATTERN))
```

AGS_IS_PATTERN_CLASS()

```
#define AGS_IS_PATTERN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_PATTERN ↩
    ))
```

AGS_PATTERN()

```
#define AGS_PATTERN(obj)                (G_TYPE_CHECK_INSTANCE_CAST(obj, AGS_TYPE_PATTERN,  ↩
    AgsPattern))
```

AGS_PATTERN_CLASS()

```
#define AGS_PATTERN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class, AGS_TYPE_PATTERN,  ↩
    AgsPatternClass))
```

AGS_PATTERN_GET_CLASS()

```
#define AGS_PATTERN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), AGS_TYPE_PATTERN ↩
    , AgsPatternClass))
```

ags_pattern_get_type ()

```
GType
ags_pattern_get_type ();
```

Types and Values

AGS_PATTERN_DEFAULT_BPM

```
#define AGS_PATTERN_DEFAULT_BPM (120.0)
```

AGS_PATTERN_TICS_PER_BEAT

```
#define AGS_PATTERN_TICS_PER_BEAT (1.0)
```

AGS_PATTERN_MINIMUM_NOTE_LENGTH

```
#define AGS_PATTERN_MINIMUM_NOTE_LENGTH (1.0 / 16.0)
```

AGS_PATTERN_MAXIMUM_NOTE_LENGTH

```
#define AGS_PATTERN_MAXIMUM_NOTE_LENGTH (16.0)
```

AGS_PATTERN_DEFAULT_LENGTH

```
#define AGS_PATTERN_DEFAULT_LENGTH (65535.0 / AGS_PATTERN_TICS_PER_BEAT -  ↩
    AGS_PATTERN_MAXIMUM_NOTE_LENGTH)
```

AGS_PATTERN_DEFAULT_JIFFIE

```
#define AGS_PATTERN_DEFAULT_JIFFIE (60.0 / AGS_PATTERN_DEFAULT_BPM /  ↩
    AGS_PATTERN_TICS_PER_BEAT)
```

AGS_PATTERN_DEFAULT_DURATION

```
#define AGS_PATTERN_DEFAULT_DURATION (AGS_PATTERN_DEFAULT_LENGTH *  ↩
    AGS_PATTERN_DEFAULT_JIFFIE * AGS_USEC_PER_SEC)
```

AGS_PATTERN_DEFAULT_OFFSET

```
#define AGS_PATTERN_DEFAULT_OFFSET (64)
```

AGS_TYPE_PATTERN

```
#define AGS_TYPE_PATTERN                (ags_pattern_get_type())
```

struct AgsPattern

```
struct AgsPattern;
```

struct AgsPatternClass

```
struct AgsPatternClass {
  GObjectClass gobject;
};
```

Property Details

The "channel" property

```
  "channel"                   AgsChannel~*
```

The pattern's channel.

Owner: AgsPattern

Flags: Read / Write

Since: 3.3.0

The "current-bit" property

```
  "current-bit"               gboolean
```

Offset of current position.

Owner: AgsPattern

Flags: Read

Default value: FALSE

Since: 3.0.0

The "first-index" property

```
"first-index"                guint
```

Selected bank 0.

Owner: AgsPattern

Flags: Read / Write

Allowed values: <= 256

Default value: 0

Since: 3.0.0

The "offset" property

```
"offset"                     guint
```

Position of pattern.

Owner: AgsPattern

Flags: Read / Write

Allowed values: <= 65535

Default value: 0

Since: 3.0.0

The "port" property

```
"port"                       AgsPort~*
```

The pattern's port.

Owner: AgsPattern

Flags: Read / Write

Since: 3.0.0

The "second-index" property

```
"second-index"               guint
```

Selected bank 1.

Owner: AgsPattern

Flags: Read / Write

Allowed values: <= 256

Default value: 0

Since: 3.0.0

The "timestamp" property

```
"timestamp"                AgsTimestamp~*
```

The pattern's timestamp.

Owner: AgsPattern

Flags: Read / Write

Since: 3.0.0

10.10 AgsPreset

AgsPreset — Presets

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PRESET_GET_OBJ_MUTEX()
GQuark	ags_preset_error_quark ()
GRecMutex *	ags_preset_get_obj_mutex ()
gboolean	ags_preset_test_flags ()
void	ags_preset_set_flags ()
void	ags_preset_unset_flags ()
GObject *	ags_preset_get_audio ()
void	ags_preset_set_audio ()
gchar *	ags_preset_get_scope ()
void	ags_preset_set_scope ()
gchar *	ags_preset_get_preset_name ()
void	ags_preset_set_preset_name ()
guint	ags_preset_get_audio_channel_start ()
void	ags_preset_set_audio_channel_start ()
guint	ags_preset_get_audio_channel_end ()
void	ags_preset_set_audio_channel_end ()
guint	ags_preset_get_pad_start ()
void	ags_preset_set_pad_start ()
guint	ags_preset_get_pad_end ()
void	ags_preset_set_pad_end ()
guint	ags_preset_get_x_start ()
void	ags_preset_set_x_start ()
guint	ags_preset_get_x_end ()
void	ags_preset_set_x_end ()
GList *	ags_preset_find_scope ()
GList *	ags_preset_find_name ()
gboolean	ags_preset_add_parameter ()
void	ags_preset_remove_parameter ()
void	ags_preset_get_parameter ()
AgsPreset *	ags_preset_new ()
#define	AGS_IS_PRESET()
#define	AGS_IS_PRESET_CLASS()

#define	AGS_PRESET()
#define	AGS_PRESET_CLASS()
#define	AGS_PRESET_GET_CLASS()
GType	ags_preset_get_type ()

Properties

GObject *	audio	Read / Write
guint	audio-channel-end	Read / Write
guint	audio-channel-start	Read / Write
guint	pad-end	Read / Write
guint	pad-start	Read / Write
char *	preset-name	Read / Write
char *	scope	Read / Write
guint	x-end	Read / Write
guint	x-start	Read / Write

Types and Values

#define	AGS_PRESET_ERROR
enum	AgsPresetError
#define	AGS_TYPE_PRESET
struct	AgsPreset
struct	AgsPresetClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsPreset
```

Includes

```
#include <ags/audio/ags_preset.h>
```

Description

The AgsPreset stores presets.

Functions

AGS_PRESET_GET_OBJ_MUTEX()

```
#define AGS_PRESET_GET_OBJ_MUTEX(obj) (&(((AgsPreset *) obj)->obj_mutex))
```

ags_preset_error_quark ()

```
GQuark
ags_preset_error_quark ();
```

ags_preset_get_obj_mutex ()

```
GRecMutex~*
ags_preset_get_obj_mutex (AgsPreset *preset);
```

Get object mutex.

Parameters

preset	the AgsPreset	

Returns

the GRecMutex to lock *preset*

Since: 3.1.0

ags_preset_test_flags ()

```
gboolean
ags_preset_test_flags (AgsPreset *preset,
                       guint flags);
```

Test *flags* to be set on *preset* .

Parameters

preset	the AgsPreset	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_preset_set_flags ()

```
void
ags_preset_set_flags (AgsPreset *preset,
                      guint flags);
```

Set flags.

Parameters

preset	the AgsPreset	
flags	the flags	

Since: 3.0.0

ags_preset_unset_flags ()

```
void
ags_preset_unset_flags (AgsPreset *preset,
                        guint flags);
```

Unset flags.

Parameters

preset	the AgsPreset	
flags	the flags	

Since: 3.0.0

ags_preset_get_audio ()

```
GObject~*
ags_preset_get_audio (AgsPreset *preset);
```

Get audio.

Parameters

preset	the AgsPreset	

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_preset_set_audio ()

```
void
ags_preset_set_audio (AgsPreset *preset,
                      GObject *audio);
```

Set audio.

Parameters

preset	the AgsPreset	
audio	the AgsAudio	

Since: 3.1.0

ags_preset_get_scope ()

```
gchar~*
ags_preset_get_scope (AgsPreset *preset);
```

Get scope.

Parameters

preset	the AgsPreset

Returns

the scope

Since: 3.1.0

ags_preset_set_scope ()

```
void
ags_preset_set_scope (AgsPreset *preset,
                      gchar *scope);
```

Set scope.

Parameters

preset	the AgsPreset
scope	the scope

Since: 3.1.0

ags_preset_get_preset_name ()

```
gchar~*
ags_preset_get_preset_name (AgsPreset *preset);
```

Get preset name.

Parameters

preset	the AgsPreset

Returns

the preset name

Since: 3.1.0

ags_preset_set_preset_name ()

```
void
ags_preset_set_preset_name (AgsPreset *preset,
                            gchar *preset_name);
```

Set preset name.

Parameters

preset	the AgsPreset	
preset_name	the preset name	

Since: 3.1.0

ags_preset_get_audio_channel_start ()

```
guint
ags_preset_get_audio_channel_start (AgsPreset *preset);
```

Gets audio channel start.

Parameters

preset	the AgsPreset	

Returns

the audio channel start

Since: 3.1.0

ags_preset_set_audio_channel_start ()

```
void
ags_preset_set_audio_channel_start (AgsPreset *preset,
                                    guint audio_channel_start);
```

Sets audio channel start.

Parameters

preset	the AgsPreset	
audio_channel_start	the audio channel start	

Since: 3.1.0

ags_preset_get_audio_channel_end ()

```
guint
ags_preset_get_audio_channel_end (AgsPreset *preset);
```

Gets audio channel end.

Parameters

preset	the AgsPreset	

Returns

the audio channel end

Since: 3.1.0

ags_preset_set_audio_channel_end ()

```
void
ags_preset_set_audio_channel_end (AgsPreset *preset,
                                  guint audio_channel_end);
```

Sets audio channel end.

Parameters

preset	the AgsPreset	
audio_channel_end	the audio channel end	

Since: 3.1.0

ags_preset_get_pad_start ()

```
guint
ags_preset_get_pad_start (AgsPreset *preset);
```

Gets pad start.

Parameters

preset	the AgsPreset	

Returns

the pad start

Since: 3.1.0

ags_preset_set_pad_start ()

```
void
ags_preset_set_pad_start (AgsPreset *preset,
                          guint pad_start);
```

Sets pad start.

Parameters

preset	the AgsPreset	
pad_start	the pad start	

Since: 3.1.0

ags_preset_get_pad_end ()

```
guint
ags_preset_get_pad_end (AgsPreset *preset);
```

Gets pad end.

Parameters

preset	the AgsPreset	

Returns

the pad end

Since: 3.1.0

ags_preset_set_pad_end ()

```
void
ags_preset_set_pad_end (AgsPreset *preset,
                        guint pad_end);
```

Sets pad end.

Parameters

preset	the AgsPreset	
pad_end	the pad end	

Since: 3.1.0

ags_preset_get_x_start ()

```
guint
ags_preset_get_x_start (AgsPreset *preset);
```

Gets x start.

Parameters

preset	the AgsPreset	

Returns

the x start

Since: 3.1.0

ags_preset_set_x_start ()

```
void
ags_preset_set_x_start (AgsPreset *preset,
                        guint x_start);
```

Sets x start.

Parameters

preset	the AgsPreset	
x_start	the x start	

Since: 3.1.0

ags_preset_get_x_end ()

```
guint
ags_preset_get_x_end (AgsPreset *preset);
```

Gets x end.

Parameters

preset	the AgsPreset	

Returns

the x end

Since: 3.1.0

ags_preset_set_x_end ()

```
void
ags_preset_set_x_end (AgsPreset *preset,
                      guint x_end);
```

Sets x end.

Parameters

preset	the AgsPreset	
x_end	the x end	

Since: 3.1.0

ags_preset_find_scope ()

```
GList~*
ags_preset_find_scope (GList *preset,
                       gchar *scope);
```

Find preset's scope in *preset* .

Parameters

preset	the GList containing AgsPreset.	*[element-type AgsAudio.Preset][transfer none]*
scope	the preset's scope	

Returns

the next matching AgsPreset.

[element-type AgsAudio.Preset][transfer none]

Since: 3.0.0

ags_preset_find_name ()

```
GList~*
ags_preset_find_name (GList *preset,
                      gchar *preset_name);
```

Find preset name in *preset* .

Parameters

preset	the GList containing AgsPreset.	*[element-type AgsAudio.Preset][transfer none]*
preset_name	the preset's name	

Returns

the next matching AgsPreset.

[element-type AgsAudio.Preset][transfer none]

Since: 3.0.0

ags_preset_add_parameter ()

```
gboolean
ags_preset_add_parameter (AgsPreset *preset,
                          gchar *param_name,
                          GValue *value);
```

Add parameter to *preset* .

Parameters

preset	the AgsPreset	
param_name	the parameter name	
value	the value to add	

Since: 3.0.0

ags_preset_remove_parameter ()

```
void
ags_preset_remove_parameter (AgsPreset *preset,
                             guint nth);
```

Remove parameter of *preset* .

Parameters

preset	the AgsPreset	
nth	the nth parameter to remove	

Since: 3.0.0

ags_preset_get_parameter ()

```
void
ags_preset_get_parameter (AgsPreset *preset,
                          gchar *param_name,
                          GValue *value,
                          GError **error);
```

Get parameter specified by *param_name* . If parameter not available the *error* is set to indicate the failure.

Parameters

preset	the AgsPreset	
param_name	the parameter name	
value	the return location of value	
error	the GError	

Since: 3.0.0

ags_preset_new ()

```
AgsPreset~*
ags_preset_new ();
```

Create a new instance of AgsPreset

Returns

the new AgsPreset

Since: 3.0.0

AGS_IS_PRESET()

```
#define AGS_IS_PRESET(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_PRESET))
```

AGS_IS_PRESET_CLASS()

```
#define AGS_IS_PRESET_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_PRESET))
```

AGS_PRESET()

```
#define AGS_PRESET(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_PRESET, ↩
    AgsPreset))
```

AGS_PRESET_CLASS()

```
#define AGS_PRESET_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_PRESET, ↩
    AgsPresetClass))
```

AGS_PRESET_GET_CLASS()

```
#define AGS_PRESET_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_PRESET, ↩
    AgsPresetClass))
```

ags_preset_get_type ()

```
GType
ags_preset_get_type ();
```

Types and Values

AGS_PRESET_ERROR

```
#define AGS_PRESET_ERROR (ags_preset_error_quark())
```

enum AgsPresetError

Members

AGS_PRESET_ERROR_NO_SUCH_PARAMETER	

AGS_TYPE_PRESET

```
#define AGS_TYPE_PRESET                 (ags_preset_get_type())
```

struct AgsPreset

```
struct AgsPreset;
```

struct AgsPresetClass

```
struct AgsPresetClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                     GObject~*
```

The AgsAudio belonging to.

Owner: AgsPreset

Flags: Read / Write

Since: 3.0.0

The "audio-channel-end" property

```
  "audio-channel-end"         guint
```

The end audio channel to apply.

Owner: AgsPreset

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "audio-channel-start" property

```
  "audio-channel-start"       guint
```

The start audio channel to apply.

Owner: AgsPreset

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "pad-end" property

```
"pad-end"                    guint
```

The end pad to apply.

Owner: AgsPreset

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "pad-start" property

```
"pad-start"                  guint
```

The start pad to apply.

Owner: AgsPreset

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "preset-name" property

```
"preset-name"                char~*
```

The preset name.

Owner: AgsPreset

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "scope" property

```
"scope"                      char~*
```

The preset's scope.

Owner: AgsPreset

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "x-end" property

```
"x-end"                      guint
```

The end x to apply.

Owner: AgsPreset

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "x-start" property

```
"x-start"                    guint
```

The start x to apply.

Owner: AgsPreset

Flags: Read / Write

Default value: 0

Since: 3.0.0

Chapter 11

Fundamental recall types

The fundamental recall types are related to effect processing.

11.1 AgsFxFactory

AgsFxFactory — Factory pattern

Stability Level

Stable, unless otherwise indicated

Functions

GList *	ags_fx_factory_create ()
#define	AGS_FX_FACTORY()
#define	AGS_FX_FACTORY_CLASS()
#define	AGS_FX_FACTORY_GET_CLASS()
#define	AGS_IS_FX_FACTORY()
#define	AGS_IS_FX_FACTORY_CLASS()
GType	ags_fx_factory_create_flags_get_type ()
GType	ags_fx_factory_get_type ()

Types and Values

enum	AgsFxFactoryCreateFlags
#define	AGS_TYPE_FX_FACTORY
#define	AGS_TYPE_FX_FACTORY_CREATE_FLAGS
struct	AgsFxFactory
struct	AgsFxFactoryClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsFxFactoryCreateFlags
GObject
&#x2570;&#x2500;&#x2500; AgsFxFactory
```

Includes

```
#include <ags/audio/ags_fx_factory.h>
```

Description

Factory function to instantiate fx recalls.

Functions

ags_fx_factory_create ()

```
GList~*
ags_fx_factory_create (AgsAudio *audio,
                       AgsRecallContainer *play_container,
                       AgsRecallContainer *recall_container,
                       gchar *plugin_name,
                       gchar *filename,
                       gchar *effect,
                       guint start_audio_channel,
                       guint stop_audio_channel,
                       guint start_pad,
                       guint stop_pad,
                       gint position,
                       guint create_flags,
                       guint recall_flags);
```

Instantiate AgsRecall by this factory.

Parameters

audio	an AgsAudio that should keep the recalls	
play_container	an AgsRecallContainer to indetify what recall to use	
recall_container	an AgsRecallContainer to indetify what recall to use	
plugin_name	the plugin identifier to instantiate	
filename	the plugin filename	
effect	the plugin effect	
start_audio_channel	the first audio channel to apply	
stop_audio_channel	the last audio channel to apply	
start_pad	the first pad to apply	
stop_pad	the last pad to apply	
position	the position to insert the recall	
create_flags	modify the behaviour of this function	
recall_flags	flags to be set for AgsRecall	

Returns

The GList containing AgsRecall.

[element-type AgsAudio.Recall][transfer full]

Since: 3.3.0

AGS_FX_FACTORY()

```
#define AGS_FX_FACTORY(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_FACTORY, AgsFxFactory))
```

AGS_FX_FACTORY_CLASS()

```
#define AGS_FX_FACTORY_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_FACTORY, AgsFxFactoryClass))
```

AGS_FX_FACTORY_GET_CLASS()

```
#define AGS_FX_FACTORY_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_FX_FACTORY, AgsFxFactoryClass))
```

AGS_IS_FX_FACTORY()

```
#define AGS_IS_FX_FACTORY(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_FX_FACTORY))
```

AGS_IS_FX_FACTORY_CLASS()

```
#define AGS_IS_FX_FACTORY_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_FX_FACTORY))
```

ags_fx_factory_create_flags_get_type ()

```
GType
ags_fx_factory_create_flags_get_type ();
```

ags_fx_factory_get_type ()

```
GType
ags_fx_factory_get_type (void);
```

Types and Values

enum AgsFxFactoryCreateFlags

Enum values controlling instantiating the recall implementation.

Members

AGS_FX_FACTORY_OUTPUT	output related
AGS_FX_FACTORY_INPUT	input related
AGS_FX_FACTORY_REMAP	remap
AGS_FX_FACTORY_ADD	do add
AGS_FX_FACTORY_PLAY	bounded to play
AGS_FX_FACTORY_RECALL	bounded to recall
AGS_FX_FACTORY_BULK	operates on bulk mode
AGS_FX_FACTORY_LIVE	is live version

AGS_TYPE_FX_FACTORY

```
#define AGS_TYPE_FX_FACTORY                (ags_fx_factory_get_type())
```

AGS_TYPE_FX_FACTORY_CREATE_FLAGS

```
#define AGS_TYPE_FX_FACTORY_CREATE_FLAGS   (ags_fx_factory_create_flags_get_type())
```

struct AgsFxFactory

```
struct AgsFxFactory;
```

struct AgsFxFactoryClass

```
struct AgsFxFactoryClass {
  GObjectClass gobject;
};
```

11.2 AgsPlaybackDomain

AgsPlaybackDomain — Outputting audio to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PLAYBACK_DOMAIN_GET_OBJ_MUTEX()
gboolean	ags_playback_domain_test_flags ()
void	ags_playback_domain_set_flags ()
void	ags_playback_domain_unset_flags ()
void	ags_playback_domain_set_audio_thread ()
AgsThread *	ags_playback_domain_get_audio_thread ()
void	ags_playback_domain_add_playback ()
void	ags_playback_domain_insert_playback ()
void	ags_playback_domain_remove_playback ()
AgsPlaybackDomain *	ags_playback_domain_new ()
#define	AGS_IS_PLAYBACK_DOMAIN()
#define	AGS_IS_PLAYBACK_DOMAIN_CLASS()
#define	AGS_PLAYBACK_DOMAIN()
#define	AGS_PLAYBACK_DOMAIN_CLASS()
#define	AGS_PLAYBACK_DOMAIN_GET_CLASS()
GType	ags_playback_domain_get_type ()

Properties

GObject *	audio	Read / Write
gpointer	input-playback	Read / Write
gpointer	output-playback	Read / Write

Types and Values

enum	AgsPlaybackDomainFlags
#define	AGS_TYPE_PLAYBACK_DOMAIN
struct	AgsPlaybackDomain
struct	AgsPlaybackDomainClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsPlaybackDomain
```

Includes

```
#include <ags/audio/ags_playback_domain.h>
```

Description

AgsPlaybackDomain represents a domain to output. It provides you the audio processing threads per AgsSoundScope. The

The assigned AgsAudioThread calls ags_channel_recursive_run_stage() for you.

The `output-playback` and `input-playback` properties provide a GList containing AgsPlayback.

Functions

AGS_PLAYBACK_DOMAIN_GET_OBJ_MUTEX()

```
#define AGS_PLAYBACK_DOMAIN_GET_OBJ_MUTEX(obj) (&(((AgsPlaybackDomain *) obj)->obj_mutex))
```

ags_playback_domain_test_flags ()

```
gboolean
ags_playback_domain_test_flags (AgsPlaybackDomain *playback_domain,
                                guint flags);
```

Test *flags* to be set on *playback_domain* .

Parameters

playback_domain	the AgsPlaybackDomain	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_playback_domain_set_flags ()

```
void
ags_playback_domain_set_flags (AgsPlaybackDomain *playback_domain,
                               guint flags);
```

Set flags.

Parameters

playback_domain	the AgsPlaybackDomain	
flags	the flags	

Since: 3.0.0

ags_playback_domain_unset_flags ()

```
void
ags_playback_domain_unset_flags (AgsPlaybackDomain *playback_domain,
                                 guint flags);
```

Unset flags.

Parameters

playback_domain	the AgsPlaybackDomain	
flags	the flags	

Since: 3.0.0

ags_playback_domain_set_audio_thread ()

```
void
ags_playback_domain_set_audio_thread (AgsPlaybackDomain *playback_domain,
                                      AgsThread *thread,
                                      gint sound_scope);
```

Set audio thread to specified scope.

Parameters

playback_domain	the AgsPlaybackDomain	
thread	the AgsThread	
sound_scope	the thread's scope	

Since: 3.0.0

ags_playback_domain_get_audio_thread ()

```
AgsThread~*
ags_playback_domain_get_audio_thread (AgsPlaybackDomain *playback_domain,
                                      gint sound_scope);
```

Get audio thread of specified scope.

Parameters

playback_domain	the AgsPlaybackDomain	
sound_scope	the thread's scope	

Returns

the matching AgsThread or NULL.

[transfer full]

Since: 3.0.0

ags_playback_domain_add_playback ()

```
void
ags_playback_domain_add_playback (AgsPlaybackDomain *playback_domain,
                                  GObject *playback,
                                  GType channel_type);
```

Add *playback* for *channel_type* to *playback_domain* .

Parameters

playback_domain	the AgsPlaybackDomain	
playback	the AgsPlayback	
channel_type	either an AGS_TYPE_OUTPUT or AGS_TYPE_INPUT	

Since: 3.0.0

ags_playback_domain_insert_playback ()

```
void
ags_playback_domain_insert_playback (AgsPlaybackDomain *playback_domain,
                                     GObject *playback,
                                     GType channel_type,
                                     gint position);
```

Add *playback* for *channel_type* to *playback_domain* at *position* .

Parameters

playback_domain	the AgsPlaybackDomain	
playback	the AgsPlayback	
channel_type	either an AGS_TYPE_OUTPUT or AGS_TYPE_INPUT	
position	the position	

Since: 3.7.13

ags_playback_domain_remove_playback ()

```
void
ags_playback_domain_remove_playback (AgsPlaybackDomain *playback_domain,
                                     GObject *playback,
                                     GType channel_type);
```

Remove *playback* for *channel_type* of *playback_domain* .

Parameters

playback_domain	the AgsPlaybackDomain	
playback	the AgsPlayback	
channel_type	either an AGS_TYPE_OUTPUT or AGS_TYPE_INPUT	

Since: 3.0.0

ags_playback_domain_new ()

```
AgsPlaybackDomain~*
ags_playback_domain_new (GObject *audio);
```

Creates an AgsPlaybackDomain, refering to *application_context* .

Parameters

audio	the AgsAudio

Returns

a new AgsPlaybackDomain

Since: 3.0.0

AGS_IS_PLAYBACK_DOMAIN()

```
#define AGS_IS_PLAYBACK_DOMAIN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PLAYBACK_DOMAIN))
```

AGS_IS_PLAYBACK_DOMAIN_CLASS()

```
#define AGS_IS_PLAYBACK_DOMAIN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PLAYBACK_DOMAIN))
```

AGS_PLAYBACK_DOMAIN()

```
#define AGS_PLAYBACK_DOMAIN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PLAYBACK_DOMAIN, AgsPlaybackDomain))
```

AGS_PLAYBACK_DOMAIN_CLASS()

```
#define AGS_PLAYBACK_DOMAIN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_PLAYBACK_DOMAIN, AgsPlaybackDomain))
```

AGS_PLAYBACK_DOMAIN_GET_CLASS()

```
#define AGS_PLAYBACK_DOMAIN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_PLAYBACK_DOMAIN, AgsPlaybackDomainClass))
```

ags_playback_domain_get_type ()

```
GType
ags_playback_domain_get_type ();
```

Types and Values

enum AgsPlaybackDomainFlags

Enum values to control the behavior or indicate internal state of AgsPlaybackDomain by enable/disable as flags.

Members

AGS_PLAYBACK_DOMAIN_SINGLE_THREADED	single threaded
AGS_PLAYBACK_DOMAIN_SUPER_THREADED_AUDIO	super threaded au-dio

AGS_TYPE_PLAYBACK_DOMAIN

```
#define AGS_TYPE_PLAYBACK_DOMAIN                (ags_playback_domain_get_type())
```

struct AgsPlaybackDomain

```
struct AgsPlaybackDomain;
```

struct AgsPlaybackDomainClass

```
struct AgsPlaybackDomainClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                    GObject~*
```

The assigned AgsAudio.

Owner: AgsPlaybackDomain

Flags: Read / Write

Since: 3.0.0

The "input-playback" property

```
  "input-playback"           gpointer
```

The assigned input playback.

[transfer full]

Owner: AgsPlaybackDomain

Flags: Read / Write

Since: 3.0.0

The "output-playback" property

```
"output-playback"          gpointer
```

The assigned output playback.

[transfer full]

Owner: AgsPlaybackDomain

Flags: Read / Write

Since: 3.0.0

11.3 AgsPlayback

AgsPlayback — Outputting channel to soundcard

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PLAYBACK_GET_OBJ_MUTEX()
gboolean	ags_playback_test_flags ()
void	ags_playback_set_flags ()
void	ags_playback_unset_flags ()
void	ags_playback_set_channel_thread ()
AgsThread *	ags_playback_get_channel_thread ()
void	ags_playback_set_recall_id ()
AgsRecallID *	ags_playback_get_recall_id ()
AgsPlayback *	ags_playback_find_channel ()
AgsPlayback *	ags_playback_new ()
#define	AGS_IS_PLAYBACK()
#define	AGS_IS_PLAYBACK_CLASS()
#define	AGS_PLAYBACK()
#define	AGS_PLAYBACK_CLASS()
#define	AGS_PLAYBACK_GET_CLASS()
GType	ags_playback_flags_get_type ()
GType	ags_playback_get_type ()

Properties

guint	audio-channel	Read / Write
AgsChannel *	channel	Read / Write
AgsNote *	play-note	Read / Write
AgsPlaybackDomain *	playback-domain	Read / Write

Types and Values

enum	AgsPlaybackFlags
#define	AGS_TYPE_PLAYBACK
struct	AgsPlayback
struct	AgsPlaybackClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsPlaybackFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsPlayback
```

Includes

```
#include <ags/audio/ags_playback.h>
```

Description

AgsPlayback represents a context to output. It provides you the channel processing threads per AgsSoundScope.

The assigned AgsChannelThread calls ags_channel_recursive_run_stage() for you.

Functions

AGS_PLAYBACK_GET_OBJ_MUTEX()

```
#define AGS_PLAYBACK_GET_OBJ_MUTEX(obj) (&(((AgsPlayback *) obj)->obj_mutex))
```

ags_playback_test_flags ()

```
gboolean
ags_playback_test_flags (AgsPlayback *playback,
                         guint flags);
```

Test *flags* to be set on *playback* .

Parameters

playback	the AgsPlayback	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_playback_set_flags ()

```
void
ags_playback_set_flags (AgsPlayback *playback,
                        guint flags);
```

Set flags.

Parameters

playback	the AgsPlayback	
flags	the flags	

Since: 3.0.0

ags_playback_unset_flags ()

```
void
ags_playback_unset_flags (AgsPlayback *playback,
                          guint flags);
```

Unset flags.

Parameters

playback	the AgsPlayback	
flags	the flags	

Since: 3.0.0

ags_playback_set_channel_thread ()

```
void
ags_playback_set_channel_thread (AgsPlayback *playback,
                                 AgsThread *thread,
                                 gint sound_scope);
```

Set channel thread of appropriate scope.

Parameters

playback	the AgsPlayback	
thread	the AgsChannelThread	
sound_scope	the scope of the thread to set	

Since: 3.0.0

ags_playback_get_channel_thread ()

```
AgsThread~*
ags_playback_get_channel_thread (AgsPlayback *playback,
                                 gint sound_scope);
```

Get channel thread of appropriate scope.

Parameters

playback	the AgsPlayback	
sound_scope	the scope of the thread to get	

Returns

the matching AgsThread or NULL.

[transfer full]

Since: 3.0.0

ags_playback_set_recall_id ()

```
void
ags_playback_set_recall_id (AgsPlayback *playback,
                            AgsRecallID *recall_id,
                            gint sound_scope);
```

Set recall id of appropriate scope.

Parameters

playback	the AgsPlayback	
recall_id	the AgsRecallID	
sound_scope	the scope of the recall id to set	

Since: 3.0.0

ags_playback_get_recall_id ()

```
AgsRecallID~*
ags_playback_get_recall_id (AgsPlayback *playback,
                            gint sound_scope);
```

Get recall id of appropriate scope.

Parameters

playback	the AgsPlayback	
sound_scope	the scope of the recall id to get	

Returns

the matching AgsRecallID or NULL.

[transfer full]

Since: 3.0.0

ags_playback_find_channel ()

```
AgsPlayback~*
ags_playback_find_channel (GList *playback,
                           GObject *channel);
```

Find `channel` in `playback` .

Parameters

playback	the GList containing AgsPlayback.	*[element-type AgsAudio.Playback][transfer none]*
channel	the AgsChannel	

Returns

the matching playback.

[transfer none]

Since: 3.0.0

ags_playback_new ()

```
AgsPlayback~*
ags_playback_new (GObject *channel);
```

Instantiate a playback object and assign `channel` .

Parameters

channel	the AgsChannel	

Returns

the new AgsPlayback

Since: 3.0.0

AGS_IS_PLAYBACK()

```
#define AGS_IS_PLAYBACK(obj)                (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PLAYBACK))
```

AGS_IS_PLAYBACK_CLASS()

```
#define AGS_IS_PLAYBACK_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PLAYBACK))
```

AGS_PLAYBACK()

```
#define AGS_PLAYBACK(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PLAYBACK, AgsPlayback))
```

AGS_PLAYBACK_CLASS()

```
#define AGS_PLAYBACK_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST(class, AGS_TYPE_PLAYBACK, ↩
     AgsPlayback))
```

AGS_PLAYBACK_GET_CLASS()

```
#define AGS_PLAYBACK_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS(obj, AGS_TYPE_PLAYBACK, ↩
     AgsPlaybackClass))
```

ags_playback_flags_get_type ()

```
GType
ags_playback_flags_get_type ();
```

ags_playback_get_type ()

```
GType
ags_playback_get_type ();
```

Types and Values

enum AgsPlaybackFlags

Enum values to control the behavior or indicate internal state of AgsPlayback by enable/disable as flags.

Members

AGS_PLAYBACK_SINGLE_THREADED	single threaded
AGS_PLAYBACK_SUPER_THREADED_CHANNEL	super threaded channel

AGS_TYPE_PLAYBACK

```
#define AGS_TYPE_PLAYBACK                (ags_playback_get_type())
```

struct AgsPlayback

```
struct AgsPlayback;
```

struct AgsPlaybackClass

```
struct AgsPlaybackClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            guint
```

The assigned audio channel.

Owner: AgsPlayback

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "channel" property

```
  "channel"                  AgsChannel~*
```

The assigned channel.

Owner: AgsPlayback

Flags: Read / Write

Since: 3.0.0

The "play-note" property

```
"play-note"                 AgsNote~*
```

The assigned note.

Owner: AgsPlayback

Flags: Read / Write

Since: 3.0.0

The "playback-domain" property

```
"playback-domain"           AgsPlaybackDomain~*
```

The parent playback domain.

Owner: AgsPlayback

Flags: Read / Write

Since: 3.0.0

11.4 AgsRecallID

AgsRecallID — The recall id specifies run context

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RECALL_ID_GET_OBJ_MUTEX()
void	ags_recall_id_set_sound_scope ()
gboolean	ags_recall_id_check_sound_scope ()
gboolean	ags_recall_id_test_staging_flags ()
void	ags_recall_id_set_staging_flags ()
void	ags_recall_id_unset_staging_flags ()
gboolean	ags_recall_id_check_staging_flags ()
gboolean	ags_recall_id_test_state_flags ()
void	ags_recall_id_set_state_flags ()
void	ags_recall_id_unset_state_flags ()
gboolean	ags_recall_id_check_state_flags ()
AgsRecyclingContext *	ags_recall_id_get_recycling_context ()
void	ags_recall_id_set_recycling_context ()
AgsRecallID *	ags_recall_id_find_recycling_context ()
AgsRecallID *	ags_recall_id_find_parent_recycling_context ()
AgsRecallID *	ags_recall_id_new ()
#define	AGS_IS_RECALL_ID()
#define	AGS_IS_RECALL_ID_CLASS()
#define	AGS_RECALL_ID()
#define	AGS_RECALL_ID_CLASS()

#define	AGS_RECALL_ID_GET_CLASS()
GType	ags_recall_id_get_type ()

Properties

GObject *	recycling-context	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_ID
struct	AgsRecallID
struct	AgsRecallIDClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecallID
```

Includes

```
#include <ags/audio/ags_recall_id.h>
```

Description

AgsRecallID acts as dynamic context identifier. AgsAudioSignal and AgsRecall are assigned to an AgsRecallID. The recall is looking for audio signals with the very same recall id.

There is a strong relation to AgsRecyclingContext.

Functions

AGS_RECALL_ID_GET_OBJ_MUTEX()

```
#define AGS_RECALL_ID_GET_OBJ_MUTEX(obj) (&(((AgsRecallID *) obj)->obj_mutex))
```

ags_recall_id_set_sound_scope ()

```
void
ags_recall_id_set_sound_scope (AgsRecallID *recall_id,
                               gint sound_scope);
```

ags_recall_id_check_sound_scope ()

```
gboolean
ags_recall_id_check_sound_scope (AgsRecallID *recall_id,
                                 gint sound_scope);
```

Check if *sound_scope* is set for *recall_id* .

Parameters

recall_id	the AgsRecallID	
sound_scope	the sound scope to check or -1 to check all	

Returns

TRUE if sound scope matches, otherwise FALSE

Since: 3.0.0

ags_recall_id_test_staging_flags ()

```
gboolean
ags_recall_id_test_staging_flags (AgsRecallID *recall_id,
                                  guint staging_flags);
```

Test *staging_flags* to be set on *recall_id* .

Parameters

recall_id	the AgsRecallID	
staging_flags	the staging flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_id_set_staging_flags ()

```
void
ags_recall_id_set_staging_flags (AgsRecallID *recall_id,
                                 guint staging_flags);
```

Set staging flags.

Parameters

recall_id	the AgsRecallID	
staging_flags	staging flags to set	

Since: 3.0.0

ags_recall_id_unset_staging_flags ()

```
void
ags_recall_id_unset_staging_flags (AgsRecallID *recall_id,
                                   guint staging_flags);
```

Unset staging flags.

Parameters

recall_id	the AgsRecallID	
staging_flags	staging flags to unset	

Since: 3.0.0

ags_recall_id_check_staging_flags ()

```
gboolean
ags_recall_id_check_staging_flags (AgsRecallID *recall_id,
                                   guint staging_flags);
```

Check the occurence of *staging_flags* in *recall_id* .

Parameters

recall_id	the AgsRecallID	
staging_flags	staging flags to check	

Returns

TRUE if all flags matched, otherwise FALSE

Since: 3.0.0

ags_recall_id_test_state_flags ()

```
gboolean
ags_recall_id_test_state_flags (AgsRecallID *recall_id,
                                guint state_flags);
```

Test *state_flags* to be set on *recall_id* .

Parameters

recall_id	the AgsRecallID	
state_flags	the state flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_id_set_state_flags ()

```
void
ags_recall_id_set_state_flags (AgsRecallID *recall_id,
                               guint state_flags);
```

Set state flags.

Parameters

recall_id	the AgsRecallID	
state_flags	state flags to set	

Since: 3.0.0

ags_recall_id_unset_state_flags ()

```
void
ags_recall_id_unset_state_flags (AgsRecallID *recall_id,
                                 guint state_flags);
```

Unset state flags.

Parameters

recall_id	the AgsRecallID	
state_flags	state flags to unset	

Since: 3.0.0

ags_recall_id_check_state_flags ()

```
gboolean
ags_recall_id_check_state_flags (AgsRecallID *recall_id,
                                 guint state_flags);
```

Check the occurence of *state_flags* in *recall_id*.

Parameters

recall_id	the AgsRecallID	
state_flags	state flags to check	

Returns

TRUE if all flags matched, otherwise FALSE

Since: 3.0.0

ags_recall_id_get_recycling_context ()

```
AgsRecyclingContext~*
ags_recall_id_get_recycling_context (AgsRecallID *recall_id);
```

Get recycling context.

Parameters

recall_id	the AgsRecallId	

Returns

the AgsRecyclingContext.

[transfer full]

Since: 3.3.0

ags_recall_id_set_recycling_context ()

```
void
ags_recall_id_set_recycling_context (AgsRecallID *recall_id,
                                     AgsRecyclingContext *recycling_context);
```

Set recycling context.

Parameters

recall_id	the AgsRecallId	
recycling_context	the AgsRecyclingContext	

Since: 3.3.0

ags_recall_id_find_recycling_context ()

```
AgsRecallID~*
ags_recall_id_find_recycling_context (GList *recall_id,
                                      AgsRecyclingContext *recycling_context);
```

Retrieve recall id by recycling context.

Parameters

recall_id	the GList containing AgsRecallID.	*[element-type AgsAudio.RecallID][transfer none]*
recycling_context	the AgsRecyclingContext to match	

Returns

Matching AgsRecallID.

[transfer none]

Since: 3.0.0

ags_recall_id_find_parent_recycling_context ()

```
AgsRecallID~*
ags_recall_id_find_parent_recycling_context
                                (GList *recall_id,
                                 AgsRecyclingContext *parent_recycling_context);
```

Retrieve recall id by recycling context.

Parameters

recall_id	the GList containing AgsRecallID.	*[element-type AgsAudio.RecallID][transfer none]*
parent_recycling_context	the AgsRecyclingContext to match	

Returns

Matching AgsRecallID.

[transfer none]

Since: 3.0.0

ags_recall_id_new ()

```
AgsRecallID~*
ags_recall_id_new ();
```

Creates a AgsRecallID

Returns

a new AgsRecallID

Since: 3.0.0

AGS_IS_RECALL_ID()

```
#define AGS_IS_RECALL_ID(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_ID))
```

AGS_IS_RECALL_ID_CLASS()

```
#define AGS_IS_RECALL_ID_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_ID))
```

AGS_RECALL_ID()

```
#define AGS_RECALL_ID(obj)                 (G_TYPE_CHECK_INSTANCE_CAST ((obj),  ↩
    AGS_TYPE_RECALL_ID, AgsRecallID))
```

AGS_RECALL_ID_CLASS()

```
#define AGS_RECALL_ID_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST ((class),  ↩
    AGS_TYPE_RECALL_ID, AgsRecallIDClass))
```

AGS_RECALL_ID_GET_CLASS()

```
#define AGS_RECALL_ID_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_ID, AgsRecallIDClass))
```

ags_recall_id_get_type ()

```
GType
ags_recall_id_get_type (void);
```

Types and Values

AGS_TYPE_RECALL_ID

```
#define AGS_TYPE_RECALL_ID                (ags_recall_id_get_type ())
```

struct AgsRecallID

```
struct AgsRecallID;
```

struct AgsRecallIDClass

```
struct AgsRecallIDClass {
  GObjectClass gobject;
};
```

Property Details

The "recycling-context" property

```
  "recycling-context"        GObject~*
```

The dynamic run context belonging to.

Owner: AgsRecallID

Flags: Read / Write

Since: 3.0.0

11.5 AgsRecyclingContext

AgsRecyclingContext — A context of recycling acting as dynamic context

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RECYCLING_CONTEXT_GET_OBJ_MUTEX()
GList *	ags_recycling_context_find_scope ()
void	ags_recycling_context_replace ()
void	ags_recycling_context_add ()
void	ags_recycling_context_remove ()
void	ags_recycling_context_insert ()
AgsRecyclingContext *	ags_recycling_context_get_toplevel ()
gint	ags_recycling_context_find ()
gint	ags_recycling_context_find_child ()
gint	ags_recycling_context_find_parent ()
void	ags_recycling_context_add_child ()
void	ags_recycling_context_remove_child ()
GList *	ags_recycling_context_get_child_recall_id ()
AgsRecyclingContext *	ags_recycling_context_reset_recycling ()
AgsRecyclingContext *	ags_recycling_context_new ()
#define	AGS_IS_RECYCLING_CONTEXT()
#define	AGS_IS_RECYCLING_CONTEXT_CLASS()
#define	AGS_RECYCLING_CONTEXT()
#define	AGS_RECYCLING_CONTEXT_CLASS()
#define	AGS_RECYCLING_CONTEXT_GET_CLASS()
GType	ags_recycling_context_flags_get_type ()
GType	ags_recycling_context_get_type ()

Properties

gpointer	child	Read / Write
guint64	length	Read / Write
AgsRecyclingContext *	parent	Read / Write
AgsRecallID *	recall-id	Read / Write

Types and Values

enum	AgsRecyclingContextFlags
#define	AGS_TYPE_RECYCLING_CONTEXT
struct	AgsRecyclingContext
struct	AgsRecyclingContextClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsRecyclingContextFlags
GObject
&#x2570;&#x2500;&#x2500; AgsRecyclingContext
```

Includes

```
#include <ags/audio/ags_recycling_context.h>
```

Description

AgsRecyclingContext organizes AgsRecycling objects as dynamic context within nested tree.

This object is only to reference the current recycling context. This may be replaced by another recycling context as calling ags_channel_set_link().

Further it provides the `parent` property, if this is NULL your audio signal won't be recycled anymore. This means you are going to copy the recycling's audio signals to soundcard. As it is not NULL you are entering recursion.

There is a strong relation to AgsRecallID.

Functions

AGS_RECYCLING_CONTEXT_GET_OBJ_MUTEX()

```
#define AGS_RECYCLING_CONTEXT_GET_OBJ_MUTEX(obj) (&(((AgsRecyclingContext *) obj)-> ↩
    obj_mutex))
```

ags_recycling_context_find_scope ()

```
GList~*
ags_recycling_context_find_scope (GList *recycling_context,
                                  gint sound_scope);
```

Find matching *sound_scope* in *recycling_context* .

Parameters

recycling_context	the GList containing AgsRecyclingContext.	*[element-type AgsAudio.RecyclingContext][transfer none]*
sound_scope	the sound scope	

Returns

the matching GList or NULL if not found.

[element-type AgsAudio.RecyclingContext][transfer none]

Since: 3.0.0

ags_recycling_context_replace ()

```
void
ags_recycling_context_replace (AgsRecyclingContext *recycling_context,
                               AgsRecycling *recycling,
                               gint position);
```

Replaces one recycling entry in a context.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to add	
position	the index of *recycling*	

Since: 3.0.0

ags_recycling_context_add ()

```
void
ags_recycling_context_add (AgsRecyclingContext *recycling_context,
                           AgsRecycling *recycling);
```

Adds a recycling to a context.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to add	

Since: 3.0.0

ags_recycling_context_remove ()

```
void
ags_recycling_context_remove (AgsRecyclingContext *recycling_context,
                              AgsRecycling *recycling);
```

Removes a recycling in a context.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to remove	

Since: 3.0.0

ags_recycling_context_insert ()

```
void
ags_recycling_context_insert (AgsRecyclingContext *recycling_context,
                              AgsRecycling *recycling,
                              gint position);
```

Inserts a recycling to a context.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to insert	
position	the index to insert at	

Since: 3.0.0

ags_recycling_context_get_toplevel ()

```
AgsRecyclingContext~*
ags_recycling_context_get_toplevel (AgsRecyclingContext *recycling_context);
```

Iterates the tree up to highest level.

Parameters

recycling_context	the AgsRecyclingContext	

Returns

the topmost recycling context.

[transfer full]

Since: 3.0.0

ags_recycling_context_find ()

```
gint
ags_recycling_context_find (AgsRecyclingContext *recycling_context,
                            AgsRecycling *recycling);
```

Find position of recycling within array.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to look up	

Returns

recycling array index

Since: 3.0.0

ags_recycling_context_find_child ()

```
gint
ags_recycling_context_find_child (AgsRecyclingContext *recycling_context,
                                  AgsRecycling *recycling);
```

Find position of recycling within arrays.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to look up	

Returns

recycling array index

Since: 3.0.0

ags_recycling_context_find_parent ()

```
gint
ags_recycling_context_find_parent (AgsRecyclingContext *recycling_context,
                                   AgsRecycling *recycling);
```

Find position of recycling within array.

Parameters

recycling_context	the AgsRecyclingContext	
recycling	the AgsRecycling to look up	

Returns

recycling array index

Since: 3.0.0

ags_recycling_context_add_child ()

```
void
ags_recycling_context_add_child (AgsRecyclingContext *parent,
                                 AgsRecyclingContext *child);
```

Adds a recycling context as child.

Parameters

parent	the parental AgsRecyclingContext	
child	the child	

Since: 3.0.0

ags_recycling_context_remove_child ()

```
void
ags_recycling_context_remove_child (AgsRecyclingContext *parent,
                                    AgsRecyclingContext *child);
```

Removes a recycling context of its parent.

Parameters

parent	the AgsRecyclingContext	
child	the child to remove	

Since: 3.0.0

ags_recycling_context_get_child_recall_id ()

```
GList~*
ags_recycling_context_get_child_recall_id
                                (AgsRecyclingContext *recycling_context);
```

Retrieve all child recall ids.

Parameters

recycling_context	the AgsRecyclingContext	

Returns

the AgsRecallID as GList.

[element-type AgsAudio.RecallID][transfer full]

Since: 3.0.0

ags_recycling_context_reset_recycling ()

```
AgsRecyclingContext~*
ags_recycling_context_reset_recycling (AgsRecyclingContext *recycling_context,
                                       AgsRecycling *old_first_recycling,
                                       AgsRecycling *old_last_recycling,
                                       AgsRecycling *new_first_recycling,
                                       AgsRecycling *new_last_recycling);
```

Modify recycling of context.

Parameters

recycling_context	the AgsRecyclingContext	
old_first_recycling	the first recycling to replace	
old_last_recycling	the last recycling to replace	
new_first_recycling	the first recycling to insert	
new_last_recycling	the last recycling to insert	

Returns

the new AgsRecyclingContext.

[transfer full]

Since: 3.0.0

ags_recycling_context_new ()

```
AgsRecyclingContext~*
ags_recycling_context_new (guint64 length);
```

Creates a AgsRecyclingContext, boundaries are specified by *length*

Parameters

length	array dimension of context

Returns

a new AgsRecyclingContext

Since: 3.0.0

AGS_IS_RECYCLING_CONTEXT()

```
#define AGS_IS_RECYCLING_CONTEXT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECYCLING_CONTEXT))
```

AGS_IS_RECYCLING_CONTEXT_CLASS()

```
#define AGS_IS_RECYCLING_CONTEXT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECYCLING_CONTEXT))
```

AGS_RECYCLING_CONTEXT()

```
#define AGS_RECYCLING_CONTEXT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECYCLING_CONTEXT, AgsRecyclingContext))
```

AGS_RECYCLING_CONTEXT_CLASS()

```
#define AGS_RECYCLING_CONTEXT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECYCLING_CONTEXT, AgsRecyclingContextClass))
```

AGS_RECYCLING_CONTEXT_GET_CLASS()

```
#define AGS_RECYCLING_CONTEXT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECYCLING_CONTEXT, AgsRecyclingContextClass))
```

ags_recycling_context_flags_get_type ()

```
GType
ags_recycling_context_flags_get_type ();
```

ags_recycling_context_get_type ()

```
GType
ags_recycling_context_get_type ();
```

Types and Values

enum AgsRecyclingContextFlags

Enum values to control the behavior or indicate internal state of AgsRecyclingContext by enable/disable as flags.

Members

AGS_RECYCLING_CONTEXT_CHAINED_TO_OUTPUT	chained to output
AGS_RECYCLING_CONTEXT_CHAINED_TO_INPUT	chained to input

AGS_TYPE_RECYCLING_CONTEXT

```
#define AGS_TYPE_RECYCLING_CONTEXT                (ags_recycling_context_get_type())
```

struct AgsRecyclingContext

```
struct AgsRecyclingContext;
```

struct AgsRecyclingContextClass

```
struct AgsRecyclingContextClass {
  GObjectClass gobject;
};
```

Property Details

The "child" property

```
  "child"                    gpointer
```

The child recycling contexts.

[transfer full]

Owner: AgsRecyclingContext

Flags: Read / Write

Since: 3.0.0

The "length" property

```
"length"                    guint64
```

Boundary length.

Owner: AgsRecyclingContext

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "parent" property

```
"parent"                    AgsRecyclingContext~*
```

The parent recycling context within tree.

Owner: AgsRecyclingContext

Flags: Read / Write

Since: 3.0.0

The "recall-id" property

```
"recall-id"                 AgsRecallID~*
```

The assigned AgsRecallID.

Owner: AgsRecyclingContext

Flags: Read / Write

Since: 3.0.0

11.6 AgsRecallContainer

AgsRecallContainer — Container to group recalls

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RECALL_CONTAINER_GET_OBJ_MUTEX()
gboolean	ags_recall_container_test_flags ()
void	ags_recall_container_set_flags ()
void	ags_recall_container_unset_flags ()
void	ags_recall_container_add ()
void	ags_recall_container_remove ()
AgsRecall *	ags_recall_container_get_recall_audio ()

GList *	ags_recall_container_get_recall_audio_run ()
GList *	ags_recall_container_get_recall_channel ()
GList *	ags_recall_container_get_recall_channel_run ()
GList *	ags_recall_container_find ()
AgsRecallContainer *	ags_recall_container_new ()
#define	AGS_IS_RECALL_CONTAINER()
#define	AGS_IS_RECALL_CONTAINER_CLASS()
#define	AGS_RECALL_CONTAINER()
#define	AGS_RECALL_CONTAINER_CLASS()
#define	AGS_RECALL_CONTAINER_GET_CLASS()
GType	ags_recall_container_flags_get_type ()
GType	ags_recall_container_get_type ()

Properties

AgsRecallAudio *	recall-audio	Read / Write
gpointer	recall-audio-run	Read / Write
GType *	recall-audio-run-type	Read / Write
GType *	recall-audio-type	Read / Write
gpointer	recall-channel	Read / Write
gpointer	recall-channel-run	Read / Write
GType *	recall-channel-run-type	Read / Write
GType *	recall-channel-type	Read / Write

Types and Values

enum	AgsRecallContainerFlags
enum	AgsRecallContainerFindFlags
#define	AGS_TYPE_RECALL_CONTAINER
struct	AgsRecallContainer
struct	AgsRecallContainerClass

Object Hierarchy

```
GFlags
&#x2570;&#x2500;&#x2500; AgsRecallContainerFlags
GObject
&#x2570;&#x2500;&#x2500; AgsRecallContainer
```

Implemented Interfaces

AgsRecallContainer implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_container.h>
```

Description

AgsRecallContainer groups recalls of different context.

Functions

AGS_RECALL_CONTAINER_GET_OBJ_MUTEX()

```
#define AGS_RECALL_CONTAINER_GET_OBJ_MUTEX(obj) (&(((AgsRecallContainer *) obj)->obj_mutex) ↩
    )
```

ags_recall_container_test_flags ()

```
gboolean
ags_recall_container_test_flags (AgsRecallContainer *recall_container,
                                 guint flags);
```

Test *flags* to be set on *recall_container* .

Parameters

recall_container	the AgsRecallContainer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_container_set_flags ()

```
void
ags_recall_container_set_flags (AgsRecallContainer *recall_container,
                                guint flags);
```

Set flags.

Parameters

recall_container	the AgsRecallContainer	
flags	the flags	

Since: 3.0.0

ags_recall_container_unset_flags ()

```
void
ags_recall_container_unset_flags (AgsRecallContainer *recall_container,
                                  guint flags);
```

Unset flags.

Parameters

recall_container	the AgsRecallContainer	
flags	the flags	

Since: 3.0.0

ags_recall_container_add ()

```
void
ags_recall_container_add (AgsRecallContainer *recall_container,
                          AgsRecall *recall);
```

Add *recall* to *recall_container* .

Parameters

recall_container	the AgsRecallContainer	
recall	the AgsRecall	

Since: 3.0.0

ags_recall_container_remove ()

```
void
ags_recall_container_remove (AgsRecallContainer *recall_container,
                             AgsRecall *recall);
```

Remove *recall* from *recall_container* .

Parameters

recall_container	the AgsRecallContainer	
recall	the AgsRecall	

Since: 3.0.0

ags_recall_container_get_recall_audio ()

```
AgsRecall~*
ags_recall_container_get_recall_audio (AgsRecallContainer *recall_container);
```

Retrieve recall audio of container.

Parameters

recall_container	the AgsRecallContainer	

Returns

the AgsRecallAudio.

[transfer none]

Since: 3.0.0

ags_recall_container_get_recall_audio_run ()

```
GList~*
ags_recall_container_get_recall_audio_run
                                (AgsRecallContainer *recall_container);
```

Retrieve recall audio run of container.

Parameters

recall_container	the AgsRecallContainer

Returns

the AgsRecallAudioRun as list.

[element-type AgsAudio.RecallAudioRun][transfer none]

Since: 3.0.0

ags_recall_container_get_recall_channel ()

```
GList~*
ags_recall_container_get_recall_channel
                                (AgsRecallContainer *recall_container);
```

Retrieve the recall channel of container.

Parameters

recall_container	the AgsRecallContainer

Returns

the AgsRecallChannel.

[element-type AgsAudio.RecallChannel][transfer none]

Since: 3.0.0

ags_recall_container_get_recall_channel_run ()

```
GList~*
ags_recall_container_get_recall_channel_run
                                (AgsRecallContainer *recall_container);
```

Retrieve the recall channel run of container.

Parameters

recall_container	the AgsRecallContainer	

Returns

the AgsRecall.

[element-type AgsAudio.RecallChannelRun][transfer none]

Since: 3.0.0

ags_recall_container_find ()

```
GList~*
ags_recall_container_find (GList *recall_container,
                           GType type,
                           guint find_flags,
                           AgsRecallID *recall_id);
```

Finds AgsRecall for appropriate search criteria.

Parameters

recall_container	the GList containing AgsRecallContainer.	*[element-type AgsAudio.RecallContainer][transfer none]*
type	recall type	
find_flags	search mask	
recall_id	an AgsRecallID	

Returns

the matching recalls.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_container_new ()

```
AgsRecallContainer~*
ags_recall_container_new ();
```

Create a new instance of AgsRecallContainer

Returns

the new AgsRecallContainer

Since: 3.0.0

AGS_IS_RECALL_CONTAINER()

```
#define AGS_IS_RECALL_CONTAINER(obj)            (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_RECALL_CONTAINER))
```

AGS_IS_RECALL_CONTAINER_CLASS()

```
#define AGS_IS_RECALL_CONTAINER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_RECALL_CONTAINER))
```

AGS_RECALL_CONTAINER()

```
#define AGS_RECALL_CONTAINER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_CONTAINER, AgsRecallContainer))
```

AGS_RECALL_CONTAINER_CLASS()

```
#define AGS_RECALL_CONTAINER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_CONTAINER, AgsRecallContainerClass))
```

AGS_RECALL_CONTAINER_GET_CLASS()

```
#define AGS_RECALL_CONTAINER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_RECALL_CONTAINER, AgsRecallContainerClass))
```

ags_recall_container_flags_get_type ()

```
GType
ags_recall_container_flags_get_type ();
```

ags_recall_container_get_type ()

```
GType
ags_recall_container_get_type ();
```

Types and Values

enum AgsRecallContainerFlags

Enum values to control the behavior or indicate internal state of AgsRecallContainer by enable/disable as flags.

Members

AGS_RECALL_CONTAINER_PLAY	bound to play con-text

enum AgsRecallContainerFindFlags

Enum values to specify find criteria.

Members

AGS_RECALL_CONTAINER_FIND_TYPE	find type
AGS_RECALL_CONTAINER_FIND_TEMPLATE	find template
AGS_RECALL_CONTAINER_FIND_RECALL_ID	find recall id

AGS_TYPE_RECALL_CONTAINER

```
#define AGS_TYPE_RECALL_CONTAINER                (ags_recall_container_get_type())
```

struct AgsRecallContainer

```
struct AgsRecallContainer;
```

struct AgsRecallContainerClass

```
struct AgsRecallContainerClass {
  GObjectClass gobject;
};
```

Property Details

The "recall-audio" property

```
  "recall-audio"              AgsRecallAudio~*
```

The associated recall within audio context.

Owner: AgsRecallContainer

Flags: Read / Write

Since: 3.0.0

The "recall-audio-run" property

```
  "recall-audio-run"          gpointer
```

The associated recall within dynamic audio context.

[transfer full]

Owner: AgsRecallContainer

Flags: Read / Write

Since: 3.0.0

The "recall-audio-run-type" property

```
"recall-audio-run-type"    GType~*
```

The associated recall type within dynamic audio context.

Owner: AgsRecallContainer

Flags: Read / Write

Allowed values: void

Since: 3.0.0

The "recall-audio-type" property

```
"recall-audio-type"        GType~*
```

The associated recall type within audio context.

Owner: AgsRecallContainer

Flags: Read / Write

Allowed values: void

Since: 3.0.0

The "recall-channel" property

```
"recall-channel"           gpointer
```

The associated recall within channel context.

[transfer full]

Owner: AgsRecallContainer

Flags: Read / Write

Since: 3.0.0

The "recall-channel-run" property

```
"recall-channel-run"       gpointer
```

The associated recall within dynamic channel context.

[transfer full]

Owner: AgsRecallContainer

Flags: Read / Write

Since: 3.0.0

The "recall-channel-run-type" property

```
"recall-channel-run-type"  GType~*
```

The associated recall type within dynamic channel context.

Owner: AgsRecallContainer

Flags: Read / Write

Allowed values: void

Since: 3.0.0

The "recall-channel-type" property

```
"recall-channel-type"      GType~*
```

The associated recall type within channel context.

Owner: AgsRecallContainer

Flags: Read / Write

Allowed values: void

Since: 3.0.0

11.7 AgsRecallDependency

AgsRecallDependency — Recall dependency

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RECALL_DEPENDENCY_GET_OBJ_MUTEX()
GList *	ags_recall_dependency_find_dependency ()
GList *	ags_recall_dependency_find_dependency_by_provider ()
GObject *	ags_recall_dependency_resolve ()
AgsRecallDependency *	ags_recall_dependency_new ()
#define	AGS_IS_RECALL_DEPENDENCY()
#define	AGS_IS_RECALL_DEPENDENCY_CLASS()
#define	AGS_RECALL_DEPENDENCY()
#define	AGS_RECALL_DEPENDENCY_CLASS()
#define	AGS_RECALL_DEPENDENCY_GET_CLASS()
GType	ags_recall_dependency_get_type ()

Properties

AgsRecall *	dependency	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_DEPENDENCY
struct	AgsRecallDependency
struct	AgsRecallDependencyClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecallDependency
```

Includes

```
#include <ags/audio/ags_recall_dependency.h>
```

Description

AgsRecallDependency specifies dependencies on other recalls. Dependencies are resolved during initialization.

Functions

AGS_RECALL_DEPENDENCY_GET_OBJ_MUTEX()

```
#define AGS_RECALL_DEPENDENCY_GET_OBJ_MUTEX(obj) (&(((AgsRecallDependency *) obj)-> ↩
    obj_mutex))
```

ags_recall_dependency_find_dependency ()

```
GList~*
ags_recall_dependency_find_dependency (GList *recall_dependency,
                                       GObject *dependency);
```

Retrieve dependency.

Parameters

recall_dependency	the GList containing AgsRecallDependency.	*[element-type AgsAudio.RecallDependency][transfer none]*
dependency	the AgsRecall depending on	

Returns

next matching GList or NULL.

[element-type AgsAudio.RecallDependency][transfer none]

Since: 3.0.0

ags_recall_dependency_find_dependency_by_provider ()

```
GList~*
ags_recall_dependency_find_dependency_by_provider
                                (GList *recall_dependency,
                                 GObject *provider);
```

Retrieve dependency by provider.

Parameters

recall_dependency	the GList containing AgsRecallDependency.	*[element-type AgsAudio.RecallDependency][transfer none]*
provider	the object providing recall, like AgsAudio or AgsChannel	

Returns

next matching GList or NULL.

[element-type AgsAudio.RecallDependency][transfer none]

Since: 3.0.0

ags_recall_dependency_resolve ()

```
GObject~*
ags_recall_dependency_resolve (AgsRecallDependency *recall_dependency,
                               AgsRecallID *recall_id);
```

Resolve dependency.

Parameters

recall_dependency	an AgsRecallDependency	
recall_id	the AgsRecallID refering to	

Returns

the AgsRecall dependency.

[transfer none]

Since: 3.0.0

ags_recall_dependency_new ()

```
AgsRecallDependency~*
ags_recall_dependency_new (GObject *dependency);
```

Creates an AgsRecallDependency

Parameters

dependency	the AgsRecall depending on	

Returns

the new AgsRecallDependency

Since: 3.0.0

AGS_IS_RECALL_DEPENDENCY()

```
#define AGS_IS_RECALL_DEPENDENCY(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_RECALL_DEPENDENCY))
```

AGS_IS_RECALL_DEPENDENCY_CLASS()

```
#define AGS_IS_RECALL_DEPENDENCY_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_RECALL_DEPENDENCY))
```

AGS_RECALL_DEPENDENCY()

```
#define AGS_RECALL_DEPENDENCY(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_DEPENDENCY, AgsRecallDependency))
```

AGS_RECALL_DEPENDENCY_CLASS()

```
#define AGS_RECALL_DEPENDENCY_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_DEPENDENCY, AgsRecallDependencyClass))
```

AGS_RECALL_DEPENDENCY_GET_CLASS()

```
#define AGS_RECALL_DEPENDENCY_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_RECALL_DEPENDENCY, AgsRecallDependencyClass))
```

ags_recall_dependency_get_type ()

```
GType
ags_recall_dependency_get_type (void);
```

Types and Values

AGS_TYPE_RECALL_DEPENDENCY

```
#define AGS_TYPE_RECALL_DEPENDENCY                 (ags_recall_dependency_get_type())
```

struct AgsRecallDependency

```
struct AgsRecallDependency;
```

struct AgsRecallDependencyClass

```
struct AgsRecallDependencyClass {
  GObjectClass gobject;
};
```

Property Details

The "dependency" property

```
"dependency"                AgsRecall~*
```

The dependency.

Owner: AgsRecallDependency

Flags: Read / Write

Since: 3.0.0

11.8 AgsRecall

AgsRecall — The recall base class

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RECALL_HANDLER()
#define	AGS_RECALL_GET_OBJ_MUTEX()
void	ags_recall_global_set_omit_event ()
gboolean	ags_recall_global_get_children_lock_free ()
gboolean	ags_recall_global_get_omit_event ()
gboolean	ags_recall_global_get_performance_mode ()
gboolean	ags_recall_global_get_rt_safe ()
GRecMutex *	ags_recall_get_obj_mutex ()
gboolean	ags_recall_test_flags ()
void	ags_recall_set_flags ()
void	ags_recall_unset_flags ()
gboolean	ags_recall_test_ability_flags ()
void	ags_recall_set_ability_flags ()
void	ags_recall_unset_ability_flags ()
gboolean	ags_recall_check_ability_flags ()
gboolean	ags_recall_match_ability_flags_to_scope ()
gboolean	ags_recall_test_behaviour_flags ()
void	ags_recall_set_behaviour_flags ()
void	ags_recall_unset_behaviour_flags ()
gboolean	ags_recall_check_behaviour_flags ()
void	ags_recall_set_sound_scope ()
gint	ags_recall_get_sound_scope ()
gboolean	ags_recall_check_sound_scope ()
gboolean	ags_recall_test_staging_flags ()
void	ags_recall_set_staging_flags ()
void	ags_recall_unset_staging_flags ()
gboolean	ags_recall_check_staging_flags ()
gboolean	ags_recall_test_state_flags ()
void	ags_recall_set_state_flags ()
void	ags_recall_unset_state_flags ()

gboolean	ags_recall_check_state_flags ()
gchar *	ags_recall_get_filename ()
void	ags_recall_set_filename ()
gchar *	ags_recall_get_effect ()
void	ags_recall_set_effect ()
guint	ags_recall_get_effect_index ()
void	ags_recall_set_effect_index ()
GObject *	ags_recall_get_recall_container ()
void	ags_recall_set_recall_container ()
AgsRecallID *	ags_recall_get_recall_id ()
void	ags_recall_set_recall_id ()
GList *	ags_recall_get_recall_dependency ()
void	ags_recall_set_recall_dependency ()
void	ags_recall_add_recall_dependency ()
void	ags_recall_remove_recall_dependency ()
GList *	ags_recall_get_port ()
void	ags_recall_set_port ()
void	ags_recall_add_port ()
void	ags_recall_remove_port ()
GList *	ags_recall_get_children ()
void	ags_recall_set_children ()
void	ags_recall_add_child ()
void	ags_recall_remove_child ()
void	ags_recall_handler_free ()
AgsRecallHandler *	ags_recall_handler_alloc ()
void	ags_recall_add_recall_handler ()
void	ags_recall_remove_recall_handler ()
GObject *	ags_recall_get_output_soundcard ()
void	ags_recall_set_output_soundcard ()
GObject *	ags_recall_get_input_soundcard ()
void	ags_recall_set_input_soundcard ()
guint	ags_recall_get_samplerate ()
void	ags_recall_set_samplerate ()
guint	ags_recall_get_buffer_size ()
void	ags_recall_set_buffer_size ()
guint	ags_recall_get_format ()
void	ags_recall_set_format ()
void	ags_recall_resolve_dependency ()
void	ags_recall_check_rt_data ()
void	ags_recall_run_init_pre ()
void	ags_recall_run_init_inter ()
void	ags_recall_run_init_post ()
void	ags_recall_feed_input_queue ()
void	ags_recall_automate ()
void	ags_recall_run_pre ()
void	ags_recall_run_inter ()
void	ags_recall_run_post ()
void	ags_recall_do_feedback ()
void	ags_recall_feed_output_queue ()
void	ags_recall_stop_persistent ()
void	ags_recall_cancel ()
void	ags_recall_done ()
AgsRecall *	ags_recall_duplicate ()
void	ags_recall_notify_dependency ()
void	ags_recall_child_added ()
gboolean	ags_recall_is_done ()

GList *	ags_recall_get_by_effect ()
GList *	ags_recall_find_recall_id_with_effect ()
GList *	ags_recall_find_type ()
GList *	ags_recall_find_template ()
GList *	ags_recall_template_find_type ()
GList *	ags_recall_template_find_all_type ()
GList *	ags_recall_find_type_with_recycling_context ()
GList *	ags_recall_find_recycling_context ()
GList *	ags_recall_find_provider ()
GList *	ags_recall_template_find_provider ()
GList *	ags_recall_find_provider_with_recycling_context ()
void	ags_recall_lock_port ()
void	ags_recall_unlock_port ()
AgsRecall *	ags_recall_new ()
#define	AGS_IS_RECALL()
#define	AGS_IS_RECALL_CLASS()
#define	AGS_RECALL()
#define	AGS_RECALL_CLASS()
#define	AGS_RECALL_GET_CLASS()
GType	ags_recall_flags_get_type ()
GType	ags_recall_get_type ()
GType	ags_recall_notify_dependency_mode_get_type ()

Properties

guint	audio-channel	Read / Write
gpointer	automation-port	Read / Write
guint	buffer-size	Read / Write
gpointer	child	Read / Write
GType *	child-type	Read / Write
char *	effect	Read / Write
guint	effect-index	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
GObject *	input-soundcard	Read / Write
int	input-soundcard-channel	Read / Write
guint	line	Read / Write
GObject *	output-soundcard	Read / Write
int	output-soundcard-channel	Read / Write
guint	pad	Read / Write
AgsRecall *	parent	Read / Write
gpointer	port	Read / Write
AgsRecallContainer *	recall-container	Read / Write
gpointer	recall-dependency	Read / Write
AgsRecallID *	recall-id	Read / Write
guint	samplerate	Read / Write

Signals

void	automate	Run Last
void	cancel	Run Last
void	check-rt-data	Run Last
void	child-added	Run Last
void	do-feedback	Run Last

void	done	Run Last
GObject*	duplicate	Run Last
void	feed-input-queue	Run Last
void	feed-output-queue	Run Last
void	notify-dependency	Run Last
void	resolve-dependency	Run Last
void	run-init-inter	Run Last
void	run-init-post	Run Last
void	run-init-pre	Run Last
void	run-inter	Run Last
void	run-post	Run Last
void	run-pre	Run Last
void	stop-persistent	Run Last

Types and Values

#define	AGS_RECALL_DEFAULT_VERSION
#define	AGS_RECALL_DEFAULT_BUILD_ID
enum	AgsRecallFlags
enum	AgsRecallNotifyDependencyMode
struct	AgsRecallHandler
#define	AGS_TYPE_RECALL
#define	AGS_TYPE_RECALL_FLAGS
#define	AGS_TYPE_RECALL_NOTIFY_DEPENDENCY_MODE
struct	AgsRecall
struct	AgsRecallClass

Object Hierarchy

```
    GFlags
    &#x251c;&#x2500;&#x2500; AgsRecallFlags
    &#x2570;&#x2500;&#x2500; AgsRecallNotifyDependencyMode
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x251c;&#x2500;&#x2500; AgsRecallAudio
        &#x251c;&#x2500;&#x2500; AgsRecallAudioRun
        &#x251c;&#x2500;&#x2500; AgsRecallAudioSignal
        &#x251c;&#x2500;&#x2500; AgsRecallChannel
        &#x251c;&#x2500;&#x2500; AgsRecallChannelRun
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
```

Implemented Interfaces

AgsRecall implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall.h>
```

Description

AgsRecall acts as effect processor. It depends on staging program from AgsAudioThread and AgsChannelThread what signals are invoked.

ags-fx staging does actually only run duplicate and intialization of a recall:

- play-duplicate
- play-run-init-pre, play-run-init-inter and play-run-init-post

And the actuall audio processing:

- play-automate
- play-run-inter

Termination of audio processing usually follows:

- play-done
- play-cancel

You usually create recalls of the same audio processing effect for different contices. For those contices exists derived recalls you inherit from.

- AgsRecallAudio
- AgsRecallAudioRun
- AgsRecallChannel
- AgsRecallChannelRun
- AgsRecallRecycling
- AgsREcallAudioSignal

You add the derived AgsRecallAudio and AgsRecallAudioRun to AgsAudio by calling ags_audio_add_recall(). Analogously, you do this for AgsRecallChannel and AgsRecallChannelRun by calling ags_channel_add_recall().

Make sure you set property `child-type` of AgsRecallChannelRun to AgsRecallRecycling and of AgsRecallRecycling to AgsRecallAudioSignal derived type.

Further AgsRecall is packed by AgsRecallContainer.

Functions

AGS_RECALL_HANDLER()

```
#define AGS_RECALL_HANDLER(handler)     ((AgsRecallHandler *)(handler))
```

AGS_RECALL_GET_OBJ_MUTEX()

```
#define AGS_RECALL_GET_OBJ_MUTEX(obj)  (&(((AgsRecall *) obj)->obj_mutex))
```

ags_recall_global_set_omit_event ()

```
void
ags_recall_global_set_omit_event (gboolean omit_event);
```

Set global config value omit event.

Parameters

omit_event	TRUE if omit event, otherwise FALSE

Since: 3.0.0

ags_recall_global_get_children_lock_free ()

```
gboolean
ags_recall_global_get_children_lock_free
                                ();
```

Get global config value lock free children.

Returns

if TRUE does lock free children, else not

Since: 3.0.0

ags_recall_global_get_omit_event ()

```
gboolean
ags_recall_global_get_omit_event ();
```

Get global config value omit event.

Returns

if TRUE does omit events, else not

Since: 3.0.0

ags_recall_global_get_performance_mode ()

```
gboolean
ags_recall_global_get_performance_mode
                                ();
```

Get global config value performance mode.

Returns

if TRUE does performance mode, else not

Since: 3.0.0

ags_recall_global_get_rt_safe ()

```
gboolean
ags_recall_global_get_rt_safe ();
```

Get global config value rt-safe.

Returns

if TRUE does rt-safe strategy, else not

Since: 3.0.0

ags_recall_get_obj_mutex ()

```
GRecMutex~*
ags_recall_get_obj_mutex (AgsRecall *recall);
```

Get object mutex.

Parameters

recall	the AgsRecall	

Returns

the GRecMutex to lock *recall* .

[type gpointer][transfer none]

Since: 3.1.0

ags_recall_test_flags ()

```
gboolean
ags_recall_test_flags (AgsRecall *recall,
                       guint flags);
```

Test *flags* to be set on *recall* .

Parameters

recall	the AgsRecall	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_set_flags ()

```
void
ags_recall_set_flags (AgsRecall *recall,
                      guint flags);
```

Set flags.

Parameters

recall	the AgsRecall	
flags	the flags	

Since: 3.0.0

ags_recall_unset_flags ()

```
void
ags_recall_unset_flags (AgsRecall *recall,
                        guint flags);
```

Unset flags.

Parameters

recall	the AgsRecall	
flags	the flags	

Since: 3.0.0

ags_recall_test_ability_flags ()

```
gboolean
ags_recall_test_ability_flags (AgsRecall *recall,
                               guint ability_flags);
```

Test *ability_flags* to be set on *recall* .

Parameters

recall	the AgsRecall	
ability_flags	the ability flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_set_ability_flags ()

```
void
ags_recall_set_ability_flags (AgsRecall *recall,
                              guint ability_flags);
```

Set ability flags recursively.

Parameters

recall	the AgsRecall	
ability_flags	ability flags	

Since: 3.0.0

ags_recall_unset_ability_flags ()

```
void
ags_recall_unset_ability_flags (AgsRecall *recall,
                                guint ability_flags);
```

Unset ability flags recursively.

Parameters

recall	the AgsRecall	
ability_flags	ability flags	

Since: 3.0.0

ags_recall_check_ability_flags ()

```
gboolean
ags_recall_check_ability_flags (AgsRecall *recall,
                                guint ability_flags);
```

Check if *ability_flags* is set for *recall* .

Parameters

recall	the AgsRecall	
ability_flags	the ability flags	

Returns

TRUE flags are set, otherwise FALSE

Since: 3.0.0

ags_recall_match_ability_flags_to_scope ()

```
gboolean
ags_recall_match_ability_flags_to_scope
                                (AgsRecall *recall,
                                 gint sound_scope);
```

Check if *sound_scope* related ability flag is set.

Parameters

recall	the AgsRecall	
sound_scope	the sound scope	

Returns

TRUE if sound scope is available, otherwise FALSE

Since: 3.0.0

ags_recall_test_behaviour_flags ()

```
gboolean
ags_recall_test_behaviour_flags (AgsRecall *recall,
                                 guint behaviour_flags);
```

Test *behaviour_flags* to be set on *recall* .

Parameters

recall	the AgsRecall	
behaviour_flags	the behaviour flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_set_behaviour_flags ()

```
void
ags_recall_set_behaviour_flags (AgsRecall *recall,
                                guint behaviour_flags);
```

Set behaviour flags of *recall* .

Parameters

recall	the AgsRecall	
behaviour_flags	the behaviour flags	

Since: 3.0.0

ags_recall_unset_behaviour_flags ()

```
void
ags_recall_unset_behaviour_flags (AgsRecall *recall,
                                  guint behaviour_flags);
```

Unset behaviour flags of *recall* .

Parameters

recall	the AgsRecall	
behaviour_flags	the behaviour flags	

Since: 3.0.0

ags_recall_check_behaviour_flags ()

```
gboolean
ags_recall_check_behaviour_flags (AgsRecall *recall,
                                  guint behaviour_flags);
```

Check if *behaviour_flags* is set for *recall* .

Parameters

recall	the AgsRecall	
behaviour_flags	the behaviour flags	

Returns

TRUE flags are set, otherwise FALSE

Since: 3.0.0

ags_recall_set_sound_scope ()

```
void
ags_recall_set_sound_scope (AgsRecall *recall,
                            gint sound_scope);
```

Set *sound_scope* for *recall* .

Parameters

recall	the AgsRecall	
sound_scope	the sound scope	

Since: 3.0.0

ags_recall_get_sound_scope ()

```
gint
ags_recall_get_sound_scope (AgsRecall *recall);
```

Get sound scope for *recall* .

Parameters

recall	the AgsRecall	

Returns

the used sound scope

Since: 3.0.0

ags_recall_check_sound_scope ()

```
gboolean
ags_recall_check_sound_scope (AgsRecall *recall,
                              gint sound_scope);
```

Check if *sound_scope* is set for *recall* .

Parameters

recall	the AgsRecall	
sound_scope	the sound scope to check or -1 to check all	

Returns

TRUE if sound scope matches, otherwise FALSE

Since: 3.0.0

ags_recall_test_staging_flags ()

```
gboolean
ags_recall_test_staging_flags (AgsRecall *recall,
                               guint staging_flags);
```

Test *staging_flags* to be set on *recall* .

Parameters

recall	the AgsRecall	
staging_flags	the staging flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_set_staging_flags ()

```
void
ags_recall_set_staging_flags (AgsRecall *recall,
                              guint staging_flags);
```

Set staging flags.

Parameters

recall	the AgsRecall	
staging_flags	staging flags to set	

Since: 3.0.0

ags_recall_unset_staging_flags ()

```
void
ags_recall_unset_staging_flags (AgsRecall *recall,
                                guint staging_flags);
```

Unset staging flags.

Parameters

recall	the AgsRecall	
staging_flags	staging flags to unset	

Since: 3.0.0

ags_recall_check_staging_flags ()

```
gboolean
ags_recall_check_staging_flags (AgsRecall *recall,
                                guint staging_flags);
```

Check the occurence of *staging_flags* in *recall* .

Parameters

recall	the AgsRecall	
staging_flags	staging flags to check	

Returns

TRUE if all flags matched, otherwise FALSE

Since: 3.0.0

ags_recall_test_state_flags ()

```
gboolean
ags_recall_test_state_flags (AgsRecall *recall,
                             guint state_flags);
```

Test *state_flags* to be set on *recall* .

Parameters

recall	the AgsRecall	
state_flags	the state flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_recall_set_state_flags ()

```
void
ags_recall_set_state_flags (AgsRecall *recall,
                            guint state_flags);
```

Set state flags.

Parameters

recall	the AgsRecall	
state_flags	state flags to set	

Since: 3.0.0

ags_recall_unset_state_flags ()

```
void
ags_recall_unset_state_flags (AgsRecall *recall,
                              guint state_flags);
```

Unset state flags.

Parameters

recall	the AgsRecall	
state_flags	state flags to unset	

Since: 3.0.0

ags_recall_check_state_flags ()

```
gboolean
ags_recall_check_state_flags (AgsRecall *recall,
                              guint state_flags);
```

Check the occurence of *state_flags* in *recall* .

Parameters

recall	the AgsRecall	
state_flags	state flags to check	

Returns

TRUE if all flags matched, otherwise FALSE

Since: 3.0.0

ags_recall_get_filename ()

```
gchar~*
ags_recall_get_filename (AgsRecall *recall);
```

Get filename.

Parameters

recall	the AgsRecall	

Returns

the filename

Since: 3.1.0

ags_recall_set_filename ()

```
void
ags_recall_set_filename (AgsRecall *recall,
                         gchar *filename);
```

Set filename.

Parameters

recall	the AgsRecall	
filename	the filename	

Since: 3.1.0

ags_recall_get_effect ()

```
gchar~*
ags_recall_get_effect (AgsRecall *recall);
```

Get effect.

Parameters

recall	the AgsRecall	

Returns

the effect

Since: 3.1.0

ags_recall_set_effect ()

```
void
ags_recall_set_effect (AgsRecall *recall,
                       gchar *effect);
```

Set effect.

Parameters

recall	the AgsRecall	
effect	the effect	

Since: 3.1.0

ags_recall_get_effect_index ()

```
guint
ags_recall_get_effect_index (AgsRecall *recall);
```

Get effect index.

Parameters

recall	the AgsRecall	

Returns

the effect index

Since: 3.1.0

ags_recall_set_effect_index ()

```
void
ags_recall_set_effect_index (AgsRecall *recall,
                             guint effect_index);
```

Set effect index.

Parameters

recall	the AgsRecall	
effect_index	the effect index	

Since: 3.1.0

ags_recall_get_recall_container ()

```
GObject~*
ags_recall_get_recall_container (AgsRecall *recall);
```

Get recall container of *recall* .

Parameters

recall	the AgsRecall	

Returns

the AgsRecallContainer.

[transfer full]

Since: 3.1.0

ags_recall_set_recall_container ()

```
void
ags_recall_set_recall_container (AgsRecall *recall,
                                 GObject *recall_container);
```

Set *recall_container* of *recall* .

Parameters

recall	the AgsRecall	
recall_container	the AgsRecallContainer	

Since: 3.1.0

ags_recall_get_recall_id ()

```
AgsRecallID~*
ags_recall_get_recall_id (AgsRecall *recall);
```

Get recall id of *recall* .

Parameters

recall	the AgsRecall

Returns

the AgsRecallID.

[transfer full]

Since: 3.1.0

ags_recall_set_recall_id ()

```
void
ags_recall_set_recall_id (AgsRecall *recall,
                          AgsRecallID *recall_id);
```

Set *recall_id* of *recall* and all its children.

Parameters

recall	the AgsRecall
recall_id	the AgsRecallID to set

Since: 3.0.0

ags_recall_get_recall_dependency ()

```
GList~*
ags_recall_get_recall_dependency (AgsRecall *recall);
```

Get recall dependency.

Parameters

recall	the AgsRecall

Returns

the GList containig AgsRecallDependency.

[element-type AgsAudio.RecallDependency][transfer full]

Since: 3.1.0

ags_recall_set_recall_dependency ()

```
void
ags_recall_set_recall_dependency (AgsRecall *recall,
                                  GList *recall_dependency);
```

Set recall dependency by replacing existing.

Parameters

recall	the AgsRecall	
recall_dependency	the GList containing AgsRecallDependency.	*[element-type AgsAudio.RecallDependency][transfer full]*

Since: 3.1.0

ags_recall_add_recall_dependency ()

```
void
ags_recall_add_recall_dependency (AgsRecall *recall,
                                  AgsRecallDependency *recall_dependency);
```

Associate a new dependency for this recall.

Parameters

recall	the AgsRecall	
recall_dependency	the AgsRecallDependency	

Since: 3.0.0

ags_recall_remove_recall_dependency ()

```
void
ags_recall_remove_recall_dependency (AgsRecall *recall,
                                     AgsRecallDependency *recall_dependency);
```

Remove a prior associated dependency.

Parameters

recall	the AgsRecall	
recall_dependency	the AgsRecallDependency	

Since: 3.0.0

ags_recall_get_port ()

```
GList~*
ags_recall_get_port (AgsRecall *recall);
```

Get port.

Parameters

recall	the AgsRecall	

Returns

the GList containig AgsPort.

[element-type AgsAudio.Port][transfer full]

Since: 3.7.18

ags_recall_set_port ()

```
void
ags_recall_set_port (AgsRecall *recall,
                     GList *port);
```

Set port by replacing existing.

Parameters

recall	the AgsRecall	
port	the GList containing AgsPort.	*[element-type AgsAudio.Port][transfer full]*

Since: 3.7.18

ags_recall_add_port ()

```
void
ags_recall_add_port (AgsRecall *recall,
                     AgsPort *port);
```

Add *port* to *recall* .

Parameters

recall	the AgsRecall	
port	the AgsPort	

Since: 3.3.0

ags_recall_remove_port ()

```
void
ags_recall_remove_port (AgsRecall *recall,
                        AgsPort *port);
```

Remove *port* from *recall* .

Parameters

recall	the AgsRecall	
port	the AgsPort	

Since: 3.3.0

ags_recall_get_children ()

```
GList~*
ags_recall_get_children (AgsRecall *recall);
```

Get recall children.

Parameters

recall	the AgsRecall	

Returns

the GList containig AgsRecall.

[element-type AgsAudio.Recall][transfer full]

Since: 3.1.0

ags_recall_set_children ()

```
void
ags_recall_set_children (AgsRecall *recall,
                         GList *children);
```

Set recall children by replacing existing.

Parameters

recall	the AgsRecall	
children	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer full]*

Since: 3.1.0

ags_recall_add_child ()

```
void
ags_recall_add_child (AgsRecall *recall,
                      AgsRecall *child);
```

Add `child` to `recall`.

Parameters

recall	the AgsRecall	
child	the child AgsRecall	

Since: 3.0.0

ags_recall_remove_child ()

```
void
ags_recall_remove_child (AgsRecall *recall,
                         AgsRecall *child);
```

Remove *child* from *recall* .

Parameters

recall	the AgsRecall	
child	the child AgsRecall	

Since: 3.0.0

ags_recall_handler_free ()

```
void
ags_recall_handler_free (AgsRecallHandler *recall_handler);
```

Free *recall_hanlder* .

Parameters

recall_handler	the AgsRecallHandler.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_recall_handler_alloc ()

```
AgsRecallHandler~*
ags_recall_handler_alloc (const gchar *signal_name,
                          GCallback callback,
                          GObject *data);
```

Allocates AgsRecallHandler.

Parameters

signal_name	signal's name to connect	
callback	the GCallback function.	*[scope call]*
data	the data to pass the callback	

Returns

the newly allocated AgsRecallHandler.

[type gpointer][transfer none]

Since: 3.0.0

ags_recall_add_recall_handler ()

```
void
ags_recall_add_recall_handler (AgsRecall *recall,
                               AgsRecallHandler *recall_handler);
```

Connect callback to *recall* specified by *recall_handler* .

Parameters

recall	the AgsRecall to connect	
recall_handler	the signal specs.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_recall_remove_recall_handler ()

```
void
ags_recall_remove_recall_handler (AgsRecall *recall,
                                  AgsRecallHandler *recall_handler);
```

Remove a AgsRecallHandler from *recall* .

Parameters

recall	the AgsRecall to connect	
recall_handler	the signal specs.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_recall_get_output_soundcard ()

```
GObject~*
ags_recall_get_output_soundcard (AgsRecall *recall);
```

Get the output soundcard object of *recall* .

Parameters

recall	the AgsRecall	

Returns

the output soundcard.

[transfer full]

Since: 3.1.0

ags_recall_set_output_soundcard ()

```
void
ags_recall_set_output_soundcard (AgsRecall *recall,
                                 GObject *output_soundcard);
```

Set output soundcard of *recall* .

Parameters

recall	the AgsRecall	
output_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_recall_get_input_soundcard ()

```
GObject~*
ags_recall_get_input_soundcard (AgsRecall *recall);
```

Get the input soundcard object of *recall* .

Parameters

recall	the AgsRecall	

Returns

the input soundcard.

[transfer full]

Since: 3.1.0

ags_recall_set_input_soundcard ()

```
void
ags_recall_set_input_soundcard (AgsRecall *recall,
                                GObject *input_soundcard);
```

Set input soundcard of *recall* .

Parameters

recall	the AgsRecall	
input_soundcard	the GObject implementing AgsSoundcard	

Since: 3.0.0

ags_recall_get_samplerate ()

```
guint
ags_recall_get_samplerate (AgsRecall *recall);
```

Gets samplerate.

Parameters

recall	the AgsRecall	

Returns

the samplerate

Since: 3.1.0

ags_recall_set_samplerate ()

```
void
ags_recall_set_samplerate (AgsRecall *recall,
                           guint samplerate);
```

Set samplerate of *recall* .

Parameters

recall	the AgsRecall	
samplerate	the samplerate	

Since: 3.0.0

ags_recall_get_buffer_size ()

```
guint
ags_recall_get_buffer_size (AgsRecall *recall);
```

Gets buffer size.

Parameters

recall	the AgsRecall	

Returns

the buffer size

Since: 3.1.0

ags_recall_set_buffer_size ()

```
void
ags_recall_set_buffer_size (AgsRecall *recall,
                            guint buffer_size);
```

Set buffer size of *recall* .

Parameters

recall	the AgsRecall	
buffer_size	the buffer size	

Since: 3.0.0

ags_recall_get_format ()

```
guint
ags_recall_get_format (AgsRecall *recall);
```

Gets format.

Parameters

recall	the AgsRecall	

Returns

the format

Since: 3.1.0

ags_recall_set_format ()

```
void
ags_recall_set_format (AgsRecall *recall,
                       guint format);
```

Set format of *recall* .

Parameters

recall	the AgsRecall	
format	the format	

Since: 3.0.0

ags_recall_resolve_dependency ()

```
void
ags_recall_resolve_dependency (AgsRecall *recall);
```

A signal indicating that the inheriting object should resolve it's dependency.

Parameters

recall	the AgsRecall	

Since: 3.0.0

ags_recall_check_rt_data ()

```
void
ags_recall_check_rt_data (AgsRecall *recall);
```

Prepare for run, this is the pre stage within the preparation.

Parameters

recall	the AgsRecall	

Since: 3.0.0

ags_recall_run_init_pre ()

```
void
ags_recall_run_init_pre (AgsRecall *recall);
```

Prepare for run, this is the pre stage within the preparation.

Parameters

recall	the AgsRecall	

Since: 3.0.0

ags_recall_run_init_inter ()

```
void
ags_recall_run_init_inter (AgsRecall *recall);
```

Prepare for run, this is the inter stage within the preparation.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_run_init_post ()

```
void
ags_recall_run_init_post (AgsRecall *recall);
```

Prepare for run, this is the post stage within the preparation.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_feed_input_queue ()

```
void
ags_recall_feed_input_queue (AgsRecall *recall);
```

Feed input queue of *recall* .

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_automate ()

```
void
ags_recall_automate (AgsRecall *recall);
```

Automate port of *recall* .

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_run_pre ()

```
void
ags_recall_run_pre (AgsRecall *recall);
```

This is the pre stage within a run.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_run_inter ()

```
void
ags_recall_run_inter (AgsRecall *recall);
```

This is the inter stage within a run.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_run_post ()

```
void
ags_recall_run_post (AgsRecall *recall);
```

This is the post stage within a run.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_do_feedback ()

```
void
ags_recall_do_feedback (AgsRecall *recall);
```

Do feedback of *recall* .

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_feed_output_queue ()

```
void
ags_recall_feed_output_queue (AgsRecall *recall);
```

Feed output queue of *recall* .

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_stop_persistent ()

```
void
ags_recall_stop_persistent (AgsRecall *recall);
```

Unsets the AGS_SOUND_BEHAVIOUR_PERSISTENT and related behaviour flags and invokes ags_recall_done().

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_cancel ()

```
void
ags_recall_cancel (AgsRecall *recall);
```

The AgsRecall doesn't want to run anymore, it aborts further execution.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_done ()

```
void
ags_recall_done (AgsRecall *recall);
```

The AgsRecall doesn't want to run anymore, it has been done its work.

Parameters

recall	the AgsRecall

Since: 3.0.0

ags_recall_duplicate ()

```
AgsRecall~*
ags_recall_duplicate (AgsRecall *recall,
                      AgsRecallID *recall_id,
                      guint *n_params,
                      gchar **parameter_name,
                      GValue *value);
```

Should duplicate an AgsRecall, so it can pass the run stages. Mainly used for creating duplicates of templates, see AGS_RECALL_TEMP

Parameters

recall	the template AgsRecAll	
recall_id	the AgsRecallID	
n_params	guint pointer to parameter count	
parameter_name	string vector containing parameter names	
value	the GValue array	

Returns

the duplicated AgsRecall.

[transfer full]

Since: 3.0.0

ags_recall_notify_dependency ()

```
void
ags_recall_notify_dependency (AgsRecall *recall,
                              guint dependency,
                              gboolean increase);
```

Notifies a recall that an other depends on it.

Parameters

recall	the AgsRecall	
dependency	the dependency to notify for, see AgsRecallNotify-DependencyMode	
increase	if TRUE increase, else if FALSE decrease	

Since: 3.0.0

ags_recall_child_added ()

```
void
ags_recall_child_added (AgsRecall *recall,
                        AgsRecall *child);
```

A signal indicating that the a child has been added.

Parameters

recall	the AgsRecall	
child	the child AgsRecall	

Since: 3.0.0

ags_recall_is_done ()

```
gboolean
ags_recall_is_done (GList *recall,
                    GObject *recycling_context);
```

Check if recall is over.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
recycling_context	the AgsRecyclingContext	

Returns

TRUE if recall is done, otherwise FALSE

Since: 3.0.0

ags_recall_get_by_effect ()

```
GList~*
ags_recall_get_by_effect (GList *recall,
                          gchar *filename,
                          gchar *effect);
```

Finds all recalls matching `filename` and `effect` .

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
filename	the filename containing `effect` or NULL	
effect	the effect name	

Returns

a GList containing AgsRecall, or NULL if not found.

[element-type AgsAudio.Recall][transfer full]

Since: 3.0.0

ags_recall_find_recall_id_with_effect ()

```
GList~*
ags_recall_find_recall_id_with_effect (GList *recall,
                                       AgsRecallID *recall_id,
                                       gchar *filename,
                                       gchar *effect);
```

Finds next matching effect name. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
recall_id	the AgsRecallID, may be NULL	
filename	the filename or NULL	
effect	the effect name	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_find_type ()

```
GList~*
ags_recall_find_type (GList *recall,
                      GType type);
```

Finds next matching recall for type. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
type	the GType	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_find_template ()

```
GList~*
ags_recall_find_template (GList *recall);
```

Finds next template, see AGS_RECALL_TEMPLATE flag. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_template_find_type ()

```
GList~*
ags_recall_template_find_type (GList *recall,
                               GType type);
```

Finds next matching recall for type which is a template, see AGS_RECALL_TEMPLATE flag. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
type	a GType	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_template_find_all_type ()

```
GList~*
ags_recall_template_find_all_type (GList *recall,
                                   ...);
```

Finds next matching recall for type which is a template, see AGS_RECALL_TEMPLATE flag. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
...	a GType	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_find_type_with_recycling_context ()

```
GList~*
ags_recall_find_type_with_recycling_context
                                (GList *recall,
                                 GType type,
                                 GObject *recycling_context);
```

Finds next matching recall for type which has *recycling_context* , see AgsRecallId for further details about AgsRecyclingContext. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
type	the GType	
recycling_context	the AgsRecyclingContext	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_find_recycling_context ()

```
GList~*
ags_recall_find_recycling_context (GList *recall,
                                   GObject *recycling_context);
```

Finds next matching recall which has *recycling_context* , see AgsRecallId for further details about AgsRecyclingContext. Intended to be used as iteration function.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
recycling_context	the AgsRecyclingContext	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_find_provider ()

```
GList~*
ags_recall_find_provider (GList *recall,
                          GObject *provider);
```

Finds next matching recall for type which has *provider* . The *provider* may be either an AgsChannel or an AgsAudio object. This function tries to find the corresponding AgsRecallChannel and AgsRecallAudio objects of a AgsRecall to find. If these recalls contains the *provider* , the function will return.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
provider	the GObject, either AgsAudio, AgsChannel, AgsRecycling or AgsAudioSignal	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_template_find_provider ()

```
GList~*
ags_recall_template_find_provider (GList *recall,
                                   GObject *provider);
```

Finds provider eg. AgsAudio or AgsChannel within *recall* containig AgsRecall.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
provider	the GObject as provider	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_find_provider_with_recycling_context ()

```
GList~*
ags_recall_find_provider_with_recycling_context
                                (GList *recall,
                                 GObject *provider,
                                 GObject *recycling_context);
```

Like ags_recall_template_find_provider() but given additionally *recycling_context* as search parameter.

Parameters

recall	the GList containing AgsRecall.	*[element-type AgsAudio.Recall][transfer none]*
provider	the GObject as provider	
recycling_context	the AgsRecyclingContext	

Returns

next matching GList, or NULL if not found.

[element-type AgsAudio.Recall][transfer none]

Since: 3.0.0

ags_recall_lock_port ()

```
void
ags_recall_lock_port (AgsRecall *recall);
```

Locks the ports.

Parameters

recall	the AgsRecall	

Since: 3.0.0

ags_recall_unlock_port ()

```
void
ags_recall_unlock_port (AgsRecall *recall);
```

Unlocks the ports.

Parameters

recall	the AgsRecall	

Since: 3.0.0

ags_recall_new ()

```
AgsRecall~*
ags_recall_new ();
```

Instantiate AgsRecall.

Returns

the new instance of AgsRecall.

Since: 3.0.0

AGS_IS_RECALL()

```
#define AGS_IS_RECALL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_RECALL))
```

AGS_IS_RECALL_CLASS()

```
#define AGS_IS_RECALL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_RECALL))
```

AGS_RECALL()

```
#define AGS_RECALL(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_RECALL,  ↩
    AgsRecall))
```

AGS_RECALL_CLASS()

```
#define AGS_RECALL_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_RECALL,  ↩
    AgsRecallClass))
```

AGS_RECALL_GET_CLASS()

```
#define AGS_RECALL_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_RECALL,  ↩
    AgsRecallClass))
```

ags_recall_flags_get_type ()

```
GType
ags_recall_flags_get_type ();
```

ags_recall_get_type ()

```
GType
ags_recall_get_type ();
```

ags_recall_notify_dependency_mode_get_type ()

```
GType
ags_recall_notify_dependency_mode_get_type
                                ();
```

Types and Values

AGS_RECALL_DEFAULT_VERSION

```
#define AGS_RECALL_DEFAULT_VERSION "2.0.0"
```

AGS_RECALL_DEFAULT_BUILD_ID

```
#define AGS_RECALL_DEFAULT_BUILD_ID "Tue Feb  6 14:27:35 UTC 2018"
```

enum AgsRecallFlags

Enum values to control the behavior or indicate internal state of AgsRecall by enable/disable as flags.

Members

AGS_RECALL_TEMPLATE	is template
AGS_RECALL_DEFAULT_TEMPLATE	is default template
AGS_RECALL_HAS_OUTPUT_PORT	has output port
AGS_RECALL_BYPASS	don't apply effect processing
AGS_RECALL_INITIAL_RUN	initial run, first attack to audio data

enum AgsRecallNotifyDependencyMode

Modes to notify of dependencies.

Members

AGS_RECALL_NOTIFY_RUN	notify dependency as calling run
AGS_RECALL_NOTIFY_AUDIO	notify dependency audio
AGS_RECALL_NOTIFY_AUDIO_RUN	notifiy dependency audio run
AGS_RECALL_NOTIFY_CHANNEL	notifiy dependency channel
AGS_RECALL_NOTIFY_CHANNEL_RUN	notifiy dependency channel run
AGS_RECALL_NOTIFY_RECALL	notifiy dependency recall

struct AgsRecallHandler

```
struct AgsRecallHandler {
  const gchar *signal_name;
  GCallback callback;
  GObject *data;
};
```

A AgsRecallHandler acts as a callback definition

Members

const gchar *signal_name;	the signal to listen
GCallback callback;	the callback to use
GObject *data;	user data to pass

AGS_TYPE_RECALL

```
#define AGS_TYPE_RECALL                (ags_recall_get_type())
```

AGS_TYPE_RECALL_FLAGS

```
#define AGS_TYPE_RECALL_FLAGS          (ags_recall_flags_get_type())
```

AGS_TYPE_RECALL_NOTIFY_DEPENDENCY_MODE

```
#define AGS_TYPE_RECALL_NOTIFY_DEPENDENCY_MODE (ags_recall_notify_dependency_mode_get_type ↩
    ())
```

struct AgsRecall

```
struct AgsRecall;
```

struct AgsRecallClass

```
struct AgsRecallClass {
  GObjectClass gobject;

  void (*resolve_dependency)(AgsRecall *recall);
  void (*check_rt_data)(AgsRecall *recall);

  void (*run_init_pre)(AgsRecall *recall);
  void (*run_init_inter)(AgsRecall *recall);
  void (*run_init_post)(AgsRecall *recall);

  void (*feed_input_queue)(AgsRecall *recall);
  void (*automate)(AgsRecall *recall);

  void (*run_pre)(AgsRecall *recall);
  void (*run_inter)(AgsRecall *recall);
  void (*run_post)(AgsRecall *recall);
```

```
  void (*do_feedback)(AgsRecall *recall);
  void (*feed_output_queue)(AgsRecall *recall);

  void (*stop_persistent)(AgsRecall *recall);
  void (*cancel)(AgsRecall *recall);
  void (*done)(AgsRecall *recall);

  AgsRecall* (*duplicate)(AgsRecall *recall,
        AgsRecallID *recall_id,
        guint *n_params, gchar **parameter_name, GValue *value);

  void (*notify_dependency)(AgsRecall *recall, guint dependency, gboolean increase);

  void (*child_added)(AgsRecall *recall, AgsRecall *child);
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            guint
```

The nth audio channel.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "automation-port" property

```
  "automation-port"          gpointer
```

The AgsPort doing automation.

[transfer full]

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "buffer-size" property

```
  "buffer-size"              guint
```

The buffer size.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "child" property

```
"child"                    gpointer
```

The child AgsRecall.

[transfer full]

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "child-type" property

```
"child-type"               GType~*
```

The type of child AgsRecall.

Owner: AgsRecall

Flags: Read / Write

Allowed values: void

Since: 3.0.0

The "effect" property

```
"effect"                   char~*
```

The plugin's effect.

Owner: AgsRecall

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "effect-index" property

```
"effect-index"             guint
```

The effect's index.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "filename" property

```
"filename"                 char~*
```

The plugin's filename.

Owner: AgsRecall

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The format.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "input-soundcard" property

```
"input-soundcard"           GObject~*
```

The assigned soundcard.

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "input-soundcard-channel" property

```
"input-soundcard-channel"   int
```

The input soundcard channel.

Owner: AgsRecall

Flags: Read / Write

Allowed values: >= -1

Default value: -1

Since: 3.0.0

The "line" property

```
"line"                      guint
```

The nth line.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "output-soundcard" property

```
"output-soundcard"          GObject~*
```

The assigned soundcard.

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "output-soundcard-channel" property

```
"output-soundcard-channel" int
```

The output soundcard channel.

Owner: AgsRecall

Flags: Read / Write

Allowed values: >= -1

Default value: -1

Since: 3.0.0

The "pad" property

```
"pad"                      guint
```

The nth pad.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "parent" property

```
"parent"                   AgsRecall~*
```

The parent AgsRecall.

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "port" property

```
"port"                     gpointer
```

The assigned AgsPort

[transfer full]

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "recall-container" property

```
"recall-container"         AgsRecallContainer~*
```

The AgsRecallContainer packed into.

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "recall-dependency" property

```
"recall-dependency"        gpointer
```

The AgsRecallDependency.

[transfer full]

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "recall-id" property

```
"recall-id"                AgsRecallID~*
```

The AgsRecallID running in.

Owner: AgsRecall

Flags: Read / Write

Since: 3.0.0

The "samplerate" property

```
"samplerate"               guint
```

The samplerate.

Owner: AgsRecall

Flags: Read / Write

Default value: 0

Since: 3.0.0

Signal Details

The "automate" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::automate signal notifies about running automation and is normally called during ::run-pre.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "cancel" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::cancel signal notifies about cancelling playback.

Parameters

recall	the AgsRecall to cancel playback	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "check-rt-data" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::check-rt-data signal notifies about initializing stage 0.

Parameters

recall	the AgsRecall to initialize	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "child-added" signal

```
void
user_function (AgsRecall *recall,
               GObject   *child,
               gpointer   user_data)
```

The ::child-added signal notifies about children added.

Parameters

recall	the AgsRecall to add the child	
child	the AgsRecall to add	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "do-feedback" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::do-feedback signal notifies about running stage 2.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "done" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::done signal notifies about stopping playback.

Parameters

recall	the AgsRecall to finish playback	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "duplicate" signal

```
GObject*
user_function (AgsRecall *recall,
               GObject   *recall_id,
               gpointer   n_params,
               gpointer   parameter_name,
               gpointer   value,
               gpointer   user_data)
```

The ::duplicate signal notifies about instantiating.

Parameters

recall	the AgsRecall to duplicate	
recall_id	the assigned AgsRecallID	
n_params	pointer to array length	
parameter_name	parameter name string vector	
value	the GValue array	
user_data	user data set when the signal handler was connected.	

Returns

the new AgsRecall instance.

[transfer full]

Flags: Run Last

Since: 3.0.0

The "feed-input-queue" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::feed-input-queue signal notifies about running feed input queue.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "feed-output-queue" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::feed-output-queue signal notifies about running feed output queue.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "notify-dependency" signal

```
void
user_function (AgsRecall *recall,
               guint      dependency,
               gboolean   increase,
               gpointer   user_data)
```

The ::notify-dependency signal notifies about dependency added.

Parameters

recall	the AgsRecall to notify	
dependency	the kind of dependency	
increase	if TRUE increase dependency count, else if FALSE decrease	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "resolve-dependency" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::resolve-dependency signal notifies about resolving dependency.

Parameters

recall	the AgsRecall to resolve	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run-init-inter" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::run-init-inter signal notifies about initializing stage 1.

Parameters

recall	the AgsRecall to initialize	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run-init-post" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::run-init-post signal notifies about initializing stage 2.

Parameters

recall	the AgsRecall to initialize	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run-init-pre" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::run-init-pre signal notifies about initializing stage 0.

Parameters

recall	the AgsRecall to initialize	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run-inter" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::run-inter signal notifies about running stage 1.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run-post" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::run-post signal notifies about running stage 2.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "run-pre" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::run-pre signal notifies about running stage 0.

Parameters

recall	the AgsRecall to play	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop-persistent" signal

```
void
user_function (AgsRecall *recall,
               gpointer   user_data)
```

The ::stop-persistent signal notifies about definitively stopping playback.

Parameters

recall	the AgsRecall stop playback	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

11.9 AgsRecallAudio

AgsRecallAudio — The recall base class of audio context

Stability Level

Stable, unless otherwise indicated

Functions

AgsAudio *	ags_recall_audio_get_audio ()
void	ags_recall_audio_set_audio ()
AgsRecallAudio *	ags_recall_audio_new ()
#define	AGS_IS_RECALL_AUDIO()
#define	AGS_IS_RECALL_AUDIO_CLASS()
#define	AGS_RECALL_AUDIO()
#define	AGS_RECALL_AUDIO_CLASS()
#define	AGS_RECALL_AUDIO_GET_CLASS()
GType	ags_recall_audio_get_type ()

Properties

AgsAudio *	audio	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_AUDIO
struct	AgsRecallAudio
struct	AgsRecallAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x251c;&#x2500;&#x2500; AgsFxAnalyseAudio
            &#x251c;&#x2500;&#x2500; AgsFxBufferAudio
            &#x251c;&#x2500;&#x2500; AgsFxNotationAudio
            &#x251c;&#x2500;&#x2500; AgsFxEnvelopeAudio
            &#x251c;&#x2500;&#x2500; AgsFxEq10Audio
            &#x251c;&#x2500;&#x2500; AgsFxHighPassAudio
            &#x251c;&#x2500;&#x2500; AgsFxLadspaAudio
            &#x251c;&#x2500;&#x2500; AgsFxLfoAudio
            &#x251c;&#x2500;&#x2500; AgsFxLowPassAudio
            &#x251c;&#x2500;&#x2500; AgsFxPatternAudio
            &#x251c;&#x2500;&#x2500; AgsFxPeakAudio
            &#x251c;&#x2500;&#x2500; AgsFxPlaybackAudio
            &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseAudio
            &#x2570;&#x2500;&#x2500; AgsFxVolumeAudio
```

Implemented Interfaces

AgsRecallAudio implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_audio.h>
```

Description

AgsRecallAudio acts as audio recall and hosts AgsPort for audio context ports.

Functions

ags_recall_audio_get_audio ()

```
AgsAudio~*
ags_recall_audio_get_audio (AgsRecallAudio *recall_audio);
```

Get audio.

Parameters

recall_audio	the AgsRecallAudio

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_recall_audio_set_audio ()

```
void
ags_recall_audio_set_audio (AgsRecallAudio *recall_audio,
                            AgsAudio *audio);
```

Set audio.

Parameters

recall_audio	the AgsRecallAudio
audio	the AgsAudio

Since: 3.1.0

ags_recall_audio_new ()

```
AgsRecallAudio~*
ags_recall_audio_new (AgsAudio *audio);
```

Creates an AgsRecallAudio.

Parameters

audio	the assigned AgsAudio

Returns

a new AgsRecallAudio.

Since: 3.0.0

AGS_IS_RECALL_AUDIO()

```
#define AGS_IS_RECALL_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_AUDIO))
```

AGS_IS_RECALL_AUDIO_CLASS()

```
#define AGS_IS_RECALL_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_AUDIO))
```

AGS_RECALL_AUDIO()

```
#define AGS_RECALL_AUDIO(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_AUDIO, AgsRecallAudio))
```

AGS_RECALL_AUDIO_CLASS()

```
#define AGS_RECALL_AUDIO_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_AUDIO, AgsRecallAudioClass))
```

AGS_RECALL_AUDIO_GET_CLASS()

```
#define AGS_RECALL_AUDIO_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_AUDIO, AgsRecallAudioClass))
```

ags_recall_audio_get_type ()

```
GType
ags_recall_audio_get_type ();
```

Types and Values

AGS_TYPE_RECALL_AUDIO

```
#define AGS_TYPE_RECALL_AUDIO                 (ags_recall_audio_get_type())
```

struct AgsRecallAudio

```
struct AgsRecallAudio;
```

struct AgsRecallAudioClass

```
struct AgsRecallAudioClass {
  AgsRecallClass recall;
};
```

Property Details

The "audio" property

```
"audio"                    AgsAudio~*
```

The assigned audio.

Owner: AgsRecallAudio

Flags: Read / Write

Since: 3.0.0

11.10 AgsRecallAudioRun

AgsRecallAudioRun — The recall base class of dynamic audio context

Stability Level

Stable, unless otherwise indicated

Functions

AgsAudio *	ags_recall_audio_run_get_audio ()
void	ags_recall_audio_run_set_audio ()
AgsRecallAudio *	ags_recall_audio_run_get_recall_audio ()
void	ags_recall_audio_run_set_recall_audio ()
AgsRecallAudioRun *	ags_recall_audio_run_new ()
#define	AGS_IS_RECALL_AUDIO_RUN()
#define	AGS_IS_RECALL_AUDIO_RUN_CLASS()
#define	AGS_RECALL_AUDIO_RUN()
#define	AGS_RECALL_AUDIO_RUN_CLASS()
#define	AGS_RECALL_AUDIO_RUN_GET_CLASS()
GType	ags_recall_audio_run_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsRecallAudio *	recall-audio	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_AUDIO_RUN
struct	AgsRecallAudioRun
struct	AgsRecallAudioRunClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x251c;&#x2500;&#x2500; AgsFxAnalyseAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxBufferAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxNotationAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxEnvelopeAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxEq10AudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxHighPassAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxLadspaAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxLfoAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxLowPassAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxPatternAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxPeakAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxPlaybackAudioProcessor
            &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseAudioProcessor
            &#x2570;&#x2500;&#x2500; AgsFxVolumeAudioProcessor
```

Implemented Interfaces

AgsRecallAudioRun implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_audio_run.h>
```

Description

AgsRecallAudioRun acts as dynamic audio recall. It does audio processing for audio context.

Functions

ags_recall_audio_run_get_audio ()

```
AgsAudio~*
ags_recall_audio_run_get_audio (AgsRecallAudioRun *recall_audio_run);
```

Get audio.

Parameters

recall_audio_run	the AgsRecallAudioRun

Returns

the AgsAudio.

[transfer full]

Since: 3.1.0

ags_recall_audio_run_set_audio ()

```
void
ags_recall_audio_run_set_audio (AgsRecallAudioRun *recall_audio_run,
                                AgsAudio *audio);
```

Set audio.

Parameters

recall_audio_run	the AgsRecallAudioRun	
audio	the AgsAudio	

Since: 3.1.0

ags_recall_audio_run_get_recall_audio ()

```
AgsRecallAudio~*
ags_recall_audio_run_get_recall_audio (AgsRecallAudioRun *recall_audio_run);
```

Get recall audio.

Parameters

recall_audio_run	the AgsRecallAudioRun	

Returns

the AgsRecallAudio.

[transfer full]

Since: 3.1.0

ags_recall_audio_run_set_recall_audio ()

```
void
ags_recall_audio_run_set_recall_audio (AgsRecallAudioRun *recall_audio_run,
                                       AgsRecallAudio *recall_audio);
```

Set recall audio.

Parameters

recall_audio_run	the AgsRecallAudioRun	
recall_audio	the AgsRecallAudio	

Since: 3.1.0

ags_recall_audio_run_new ()

```
AgsRecallAudioRun~*
ags_recall_audio_run_new ();
```

Creates an AgsRecallAudioRun.

Returns

a new AgsRecallAudioRun.

Since: 3.0.0

AGS_IS_RECALL_AUDIO_RUN()

```
#define AGS_IS_RECALL_AUDIO_RUN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_AUDIO_RUN))
```

AGS_IS_RECALL_AUDIO_RUN_CLASS()

```
#define AGS_IS_RECALL_AUDIO_RUN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_AUDIO_RUN))
```

AGS_RECALL_AUDIO_RUN()

```
#define AGS_RECALL_AUDIO_RUN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_AUDIO_RUN, AgsRecallAudioRun))
```

AGS_RECALL_AUDIO_RUN_CLASS()

```
#define AGS_RECALL_AUDIO_RUN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_AUDIO_RUN, AgsRecallAudioRunClass))
```

AGS_RECALL_AUDIO_RUN_GET_CLASS()

```
#define AGS_RECALL_AUDIO_RUN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_AUDIO_RUN, AgsRecallAudioRunClass))
```

ags_recall_audio_run_get_type ()

```
GType
ags_recall_audio_run_get_type ();
```

Types and Values

AGS_TYPE_RECALL_AUDIO_RUN

```
#define AGS_TYPE_RECALL_AUDIO_RUN                 (ags_recall_audio_run_get_type())
```

struct AgsRecallAudioRun

```
struct AgsRecallAudioRun;
```

struct AgsRecallAudioRunClass

```
struct AgsRecallAudioRunClass {
  AgsRecallClass recall;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned audio.

Owner: AgsRecallAudioRun

Flags: Read / Write

Since: 3.0.0

The "recall-audio" property

```
  "recall-audio"              AgsRecallAudio~*
```

The recall audio belonging to.

Owner: AgsRecallAudioRun

Flags: Read / Write

Since: 3.0.0

11.11 AgsRecallChannel

AgsRecallChannel — The recall base class of channel context

Stability Level

Stable, unless otherwise indicated

Functions

AgsChannel *	ags_recall_channel_get_destination ()
void	ags_recall_channel_set_destination ()
AgsChannel *	ags_recall_channel_get_source ()
void	ags_recall_channel_set_source ()
GList *	ags_recall_channel_find_channel ()

AgsRecallChannel *	ags_recall_channel_new ()
#define	AGS_IS_RECALL_CHANNEL()
#define	AGS_IS_RECALL_CHANNEL_CLASS()
#define	AGS_RECALL_CHANNEL()
#define	AGS_RECALL_CHANNEL_CLASS()
#define	AGS_RECALL_CHANNEL_GET_CLASS()
GType	ags_recall_channel_get_type ()

Properties

AgsChannel *	destination	Read / Write
AgsRecallAudio *	recall-audio	Read / Write
AgsChannel *	source	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_CHANNEL
struct	AgsRecallChannel
struct	AgsRecallChannelClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannel
        &#x251c;&#x2500;&#x2500; AgsFxAnalyseChannel
        &#x251c;&#x2500;&#x2500; AgsFxBufferChannel
        &#x251c;&#x2500;&#x2500; AgsFxNotationChannel
        &#x251c;&#x2500;&#x2500; AgsFxEnvelopeChannel
        &#x251c;&#x2500;&#x2500; AgsFxEq10Channel
        &#x251c;&#x2500;&#x2500; AgsFxHighPassChannel
        &#x251c;&#x2500;&#x2500; AgsFxLadspaChannel
        &#x251c;&#x2500;&#x2500; AgsFxLfoChannel
        &#x251c;&#x2500;&#x2500; AgsFxLowPassChannel
        &#x251c;&#x2500;&#x2500; AgsFxPatternChannel
        &#x251c;&#x2500;&#x2500; AgsFxPeakChannel
        &#x251c;&#x2500;&#x2500; AgsFxPlaybackChannel
        &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseChannel
        &#x2570;&#x2500;&#x2500; AgsFxVolumeChannel
```

Implemented Interfaces

AgsRecallChannel implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_channel.h>
```

Description

AgsRecallChannel acts as channel recall and hosts AgsPort for audio context ports.

Functions

ags_recall_channel_get_destination ()

```
AgsChannel~*
ags_recall_channel_get_destination (AgsRecallChannel *recall_channel);
```

Get destination.

Parameters

recall_channel	the AgsRecallDestination

Returns

the AgsDestination.

[transfer full]

Since: 3.1.0

ags_recall_channel_set_destination ()

```
void
ags_recall_channel_set_destination (AgsRecallChannel *recall_channel,
                                    AgsChannel *destination);
```

Set destination.

Parameters

recall_channel	the AgsRecallChannel
destination	the AgsChannel

Since: 3.1.0

ags_recall_channel_get_source ()

```
AgsChannel~*
ags_recall_channel_get_source (AgsRecallChannel *recall_channel);
```

Get source.

Parameters

recall_channel	the AgsRecallSource

Returns

the AgsSource.

[transfer full]

Since: 3.1.0

ags_recall_channel_set_source ()

```
void
ags_recall_channel_set_source (AgsRecallChannel *recall_channel,
                               AgsChannel *source);
```

Set source.

Parameters

recall_channel	the AgsRecallChannel	
source	the AgsChannel	

Since: 3.1.0

ags_recall_channel_find_channel ()

```
GList~*
ags_recall_channel_find_channel (GList *recall_channel,
                                 AgsChannel *source);
```

Retrieve next recall assigned to channel.

Parameters

recall_channel	the GList containing AgsRecallChannel.	*[element-type AgsAudio.RecallChannel][transfer none]*
source	the AgsChannel to find	

Returns

next matching GList or NULL.

[element-type AgsAudio.RecallChannel][transfer none]

Since: 3.0.0

ags_recall_channel_new ()

```
AgsRecallChannel~*
ags_recall_channel_new ();
```

Creates an AgsRecallChannel.

Returns

a new AgsRecallChannel.

Since: 3.0.0

AGS_IS_RECALL_CHANNEL()

```
#define AGS_IS_RECALL_CHANNEL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_CHANNEL))
```

AGS_IS_RECALL_CHANNEL_CLASS()

```
#define AGS_IS_RECALL_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_CHANNEL))
```

AGS_RECALL_CHANNEL()

```
#define AGS_RECALL_CHANNEL(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_CHANNEL, AgsRecallChannel))
```

AGS_RECALL_CHANNEL_CLASS()

```
#define AGS_RECALL_CHANNEL_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_CHANNEL, AgsRecallChannelClass))
```

AGS_RECALL_CHANNEL_GET_CLASS()

```
#define AGS_RECALL_CHANNEL_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_CHANNEL, AgsRecallChannelClass))
```

ags_recall_channel_get_type ()

```
GType
ags_recall_channel_get_type ();
```

Types and Values

AGS_TYPE_RECALL_CHANNEL

```
#define AGS_TYPE_RECALL_CHANNEL                 (ags_recall_channel_get_type())
```

struct AgsRecallChannel

```
struct AgsRecallChannel;
```

struct AgsRecallChannelClass

```
struct AgsRecallChannelClass {
  AgsRecallClass recall;
};
```

Property Details

The "destination" property

```
"destination"              AgsChannel~*
```

The assigned destination channel.

Owner: AgsRecallChannel

Flags: Read / Write

Since: 3.0.0

The "recall-audio" property

```
"recall-audio"             AgsRecallAudio~*
```

The associated recall within audio context.

Owner: AgsRecallChannel

Flags: Read / Write

Since: 3.0.0

The "source" property

```
"source"                   AgsChannel~*
```

The assigned source channel.

Owner: AgsRecallChannel

Flags: Read / Write

Since: 3.0.0

11.12 AgsRecallChannelRun

AgsRecallChannelRun — The recall base class of dynamic channel context

Stability Level

Stable, unless otherwise indicated

Functions

AgsRecallAudio *	ags_recall_channel_run_get_recall_audio ()
void	ags_recall_channel_run_set_recall_audio ()
AgsRecallAudioRun *	ags_recall_channel_run_get_recall_audio_run ()
void	ags_recall_channel_run_set_recall_audio_run ()
AgsRecallChannel *	ags_recall_channel_run_get_recall_channel ()
void	ags_recall_channel_run_set_recall_channel ()
AgsChannel *	ags_recall_channel_run_get_destination ()

void	ags_recall_channel_run_set_destination ()
AgsChannel *	ags_recall_channel_run_get_source ()
void	ags_recall_channel_run_set_source ()
AgsRecallChannelRun *	ags_recall_channel_run_new ()
#define	AGS_IS_RECALL_CHANNEL_RUN()
#define	AGS_IS_RECALL_CHANNEL_RUN_CLASS()
#define	AGS_RECALL_CHANNEL_RUN()
#define	AGS_RECALL_CHANNEL_RUN_CLASS()
#define	AGS_RECALL_CHANNEL_RUN_GET_CLASS()
GType	ags_recall_channel_run_get_type ()

Properties

AgsChannel *	destination	Read / Write
AgsRecallAudio *	recall-audio	Read / Write
AgsRecallAudioRun *	recall-audio-run	Read / Write
AgsRecallChannel *	recall-channel	Read / Write
AgsChannel *	source	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_CHANNEL_RUN
struct	AgsRecallChannelRun
struct	AgsRecallChannelRunClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
        &#x251c;&#x2500;&#x2500; AgsFxAnalyseChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxBufferChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxNotationChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxEnvelopeChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxEq10ChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxHighPassChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxLadspaChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxLfoChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxLowPassChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxPatternChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxPeakChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxPlaybackChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseChannelProcessor
        &#x251c;&#x2500;&#x2500; AgsFxVolumeChannelProcessor
        &#x2570;&#x2500;&#x2500; AgsGenericRecallChannelRun
```

Implemented Interfaces

AgsRecallChannelRun implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_channel_run.h>
```

Description

AgsRecallChannelRun acts as dynamic channel recall. It does audio processing for channel context.

Functions

ags_recall_channel_run_get_recall_audio ()

```
AgsRecallAudio~*
ags_recall_channel_run_get_recall_audio
                                (AgsRecallChannelRun *recall_channel_run);
```

Get recall audio.

Parameters

recall_channel_run	the AgsRecallChannelRun

Returns

the AgsRecallAudio.

[transfer full]

Since: 3.1.0

ags_recall_channel_run_set_recall_audio ()

```
void
ags_recall_channel_run_set_recall_audio
                                (AgsRecallChannelRun *recall_channel_run,
                                 AgsRecallAudio *recall_audio);
```

Set recall audio.

Parameters

recall_channel_run	the AgsRecallChannelRun
recall_audio	the AgsRecallAudio

Since: 3.1.0

ags_recall_channel_run_get_recall_audio_run ()

```
AgsRecallAudioRun~*
ags_recall_channel_run_get_recall_audio_run
                                (AgsRecallChannelRun *recall_channel_run);
```

Get recall audio.

Parameters

recall_channel_run	the AgsRecallChannelRun

Returns

the AgsRecallAudioRun.

[transfer full]

Since: 3.1.0

ags_recall_channel_run_set_recall_audio_run ()

```
void
ags_recall_channel_run_set_recall_audio_run
                                (AgsRecallChannelRun *recall_channel_run,
                                 AgsRecallAudioRun *recall_audio_run);
```

Set recall audio.

Parameters

recall_channel_run	the AgsRecallChannelRun
recall_audio_run	the AgsRecallAudioRun

Since: 3.1.0

ags_recall_channel_run_get_recall_channel ()

```
AgsRecallChannel~*
ags_recall_channel_run_get_recall_channel
                                (AgsRecallChannelRun *recall_channel_run);
```

Get recall audio.

Parameters

recall_channel_run	the AgsRecallChannelRun

Returns

the AgsRecallChannel.

[transfer full]

Since: 3.1.0

ags_recall_channel_run_set_recall_channel ()

```
void
ags_recall_channel_run_set_recall_channel
                                (AgsRecallChannelRun *recall_channel_run,
                                 AgsRecallChannel *recall_channel);
```

Set recall audio.

Parameters

recall_channel_run	the AgsRecallChannelRun	
recall_channel	the AgsRecallChannel	

Since: 3.1.0

ags_recall_channel_run_get_destination ()

```
AgsChannel~*
ags_recall_channel_run_get_destination
                                (AgsRecallChannelRun *recall_channel_run);
```

Get destination.

Parameters

recall_channel_run	the AgsRecallDestination	

Returns

the AgsDestination.

[transfer full]

Since: 3.1.0

ags_recall_channel_run_set_destination ()

```
void
ags_recall_channel_run_set_destination
                                (AgsRecallChannelRun *recall_channel_run,
                                 AgsChannel *destination);
```

Set destination.

Parameters

recall_channel_run	the AgsRecallChannelRun	
destination	the AgsChannel	

Since: 3.1.0

ags_recall_channel_run_get_source ()

```
AgsChannel~*
ags_recall_channel_run_get_source (AgsRecallChannelRun *recall_channel_run);
```

Get source.

Parameters

recall_channel_run	the AgsRecallSource

Returns

the AgsSource.

[transfer full]

Since: 3.1.0

ags_recall_channel_run_set_source ()

```
void
ags_recall_channel_run_set_source (AgsRecallChannelRun *recall_channel_run,
                                   AgsChannel *source);
```

Set source.

Parameters

recall_channel_run	the AgsRecallChannelRun
source	the AgsChannel

Since: 3.1.0

ags_recall_channel_run_new ()

```
AgsRecallChannelRun~*
ags_recall_channel_run_new ();
```

Creates an AgsRecallChannelRun

Returns

a new AgsRecallChannelRun

Since: 3.0.0

AGS_IS_RECALL_CHANNEL_RUN()

```
#define AGS_IS_RECALL_CHANNEL_RUN(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_CHANNEL_RUN))
```

AGS_IS_RECALL_CHANNEL_RUN_CLASS()

```
#define AGS_IS_RECALL_CHANNEL_RUN_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_CHANNEL_RUN))
```

AGS_RECALL_CHANNEL_RUN()

```
#define AGS_RECALL_CHANNEL_RUN(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_CHANNEL_RUN, AgsRecallChannelRun))
```

AGS_RECALL_CHANNEL_RUN_CLASS()

```
#define AGS_RECALL_CHANNEL_RUN_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_CHANNEL_RUN, AgsRecallChannelRunClass))
```

AGS_RECALL_CHANNEL_RUN_GET_CLASS()

```
#define AGS_RECALL_CHANNEL_RUN_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_CHANNEL_RUN, AgsRecallChannelRunClass))
```

ags_recall_channel_run_get_type ()

```
GType
ags_recall_channel_run_get_type ();
```

Types and Values

AGS_TYPE_RECALL_CHANNEL_RUN

```
#define AGS_TYPE_RECALL_CHANNEL_RUN                 (ags_recall_channel_run_get_type())
```

struct AgsRecallChannelRun

```
struct AgsRecallChannelRun;
```

struct AgsRecallChannelRunClass

```
struct AgsRecallChannelRunClass {
  AgsRecallClass recall;
};
```

Property Details

The "destination" property

```
  "destination"              AgsChannel~*
```

The channel to do output to.

Owner: AgsRecallChannelRun

Flags: Read / Write

Since: 3.0.0

The "recall-audio" property

```
"recall-audio"              AgsRecallAudio~*
```

The audio context of this recall.

Owner: AgsRecallChannelRun

Flags: Read / Write

Since: 3.0.0

The "recall-audio-run" property

```
"recall-audio-run"          AgsRecallAudioRun~*
```

The audio run context of this recall.

Owner: AgsRecallChannelRun

Flags: Read / Write

Since: 3.0.0

The "recall-channel" property

```
"recall-channel"            AgsRecallChannel~*
```

The channel context of this recall.

Owner: AgsRecallChannelRun

Flags: Read / Write

Since: 3.0.0

The "source" property

```
"source"                    AgsChannel~*
```

The channel to do input from.

Owner: AgsRecallChannelRun

Flags: Read / Write

Since: 3.0.0

11.13 AgsRecallRecycling

AgsRecallRecycling — The recall base class of recycling context

Stability Level

Stable, unless otherwise indicated

Functions

AgsRecallRecycling *	ags_recall_recycling_new ()
#define	AGS_IS_RECALL_RECYCLING()
#define	AGS_IS_RECALL_RECYCLING_CLASS()
#define	AGS_RECALL_RECYCLING()
#define	AGS_RECALL_RECYCLING_CLASS()
#define	AGS_RECALL_RECYCLING_GET_CLASS()
GType	ags_recall_recycling_get_type ()

Properties

guint	audio-channel	Read / Write
AgsAudioSignal *	child-destination	Read / Write
gpointer	child-source	Write
AgsRecycling *	destination	Read / Write
AgsRecycling *	source	Read / Write

Types and Values

enum	AgsRecallRecyclingFlags
#define	AGS_TYPE_RECALL_RECYCLING
struct	AgsRecallRecycling
struct	AgsRecallRecyclingClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallRecycling
        &#x251c;&#x2500;&#x2500; AgsFxAnalyseRecycling
        &#x251c;&#x2500;&#x2500; AgsFxBufferRecycling
        &#x251c;&#x2500;&#x2500; AgsFxNotationRecycling
        &#x251c;&#x2500;&#x2500; AgsFxEnvelopeRecycling
        &#x251c;&#x2500;&#x2500; AgsFxEq10Recycling
        &#x251c;&#x2500;&#x2500; AgsFxHighPassRecycling
        &#x251c;&#x2500;&#x2500; AgsFxLadspaRecycling
        &#x251c;&#x2500;&#x2500; AgsFxLfoRecycling
        &#x251c;&#x2500;&#x2500; AgsFxLowPassRecycling
        &#x251c;&#x2500;&#x2500; AgsFxPatternRecycling
        &#x251c;&#x2500;&#x2500; AgsFxPeakRecycling
        &#x251c;&#x2500;&#x2500; AgsFxPlaybackRecycling
        &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseRecycling
        &#x251c;&#x2500;&#x2500; AgsFxVolumeRecycling
        &#x2570;&#x2500;&#x2500; AgsGenericRecallRecycling
```

Implemented Interfaces

AgsRecallRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_recycling.h>
```

Description

AgsRecallRecycling acts as dynamic recycling recall. It usually does not do any audio processing but providing AgsRecallAudioSignal child recalls.

Functions

ags_recall_recycling_new ()

```
AgsRecallRecycling~*
ags_recall_recycling_new ();
```

Creates a new instance of AgsRecallRecycling

Returns

the new AgsRecallRecycling

Since: 3.0.0

AGS_IS_RECALL_RECYCLING()

```
#define AGS_IS_RECALL_RECYCLING(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_RECYCLING))
```

AGS_IS_RECALL_RECYCLING_CLASS()

```
#define AGS_IS_RECALL_RECYCLING_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_RECYCLING))
```

AGS_RECALL_RECYCLING()

```
#define AGS_RECALL_RECYCLING(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_RECYCLING, AgsRecallRecycling))
```

AGS_RECALL_RECYCLING_CLASS()

```
#define AGS_RECALL_RECYCLING_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_RECYCLING, AgsRecallRecyclingClass))
```

AGS_RECALL_RECYCLING_GET_CLASS()

```
#define AGS_RECALL_RECYCLING_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_RECYCLING, AgsRecallRecyclingClass))
```

ags_recall_recycling_get_type ()

```
GType
ags_recall_recycling_get_type ();
```

Types and Values

enum AgsRecallRecyclingFlags

Enum values to control the behavior or indicate internal state of AgsRecallRecycling by enable/disable as flags.

Members

AGS_RECALL_RECYCLING_GARBAGE_COLLECTOR	not used
AGS_RECALL_RECYCLING_MAP_CHILD_DESTINATION	not used
AGS_RECALL_RECYCLING_MAP_CHILD_SOURCE	not used
AGS_RECALL_RECYCLING_CREATE_DESTINATION_ON_MAP_SOURCE	not used

AGS_TYPE_RECALL_RECYCLING

```
#define AGS_TYPE_RECALL_RECYCLING                (ags_recall_recycling_get_type())
```

struct AgsRecallRecycling

```
struct AgsRecallRecycling;
```

struct AgsRecallRecyclingClass

```
struct AgsRecallRecyclingClass {
  AgsRecallClass recall;
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            guint
```

The assigned destination recycling.

Owner: AgsRecallRecycling

Flags: Read / Write

Allowed values: <= 65536

Default value: 0

Since: 3.0.0

The "child-destination" property

```
"child-destination"         AgsAudioSignal~*
```

The assigned destination audio signal.

Owner: AgsRecallRecycling

Flags: Read / Write

Since: 3.0.0

The "child-source" property

```
"child-source"              gpointer
```

The assigned source audio signal.

[transfer full]

Owner: AgsRecallRecycling

Flags: Write

Since: 3.0.0

The "destination" property

```
"destination"               AgsRecycling~*
```

The assigned destination recycling.

Owner: AgsRecallRecycling

Flags: Read / Write

Since: 3.0.0

The "source" property

```
"source"                    AgsRecycling~*
```

The assigned source recycling.

Owner: AgsRecallRecycling

Flags: Read / Write

Since: 3.0.0

11.14 AgsRecallAudioSignal

AgsRecallAudioSignal — The recall base class of dynamic audio signal context

Stability Level

Stable, unless otherwise indicated

Functions

AgsRecallAudioSignal *	ags_recall_audio_signal_new ()
#define	AGS_IS_RECALL_AUDIO_SIGNAL()
#define	AGS_IS_RECALL_AUDIO_SIGNAL_CLASS()
#define	AGS_RECALL_AUDIO_SIGNAL()
#define	AGS_RECALL_AUDIO_SIGNAL_CLASS()
#define	AGS_RECALL_AUDIO_SIGNAL_GET_CLASS()
GType	ags_recall_audio_signal_get_type ()

Properties

AgsAudioSignal *	destination	Read / Write
AgsAudioSignal *	source	Read / Write

Types and Values

#define	AGS_TYPE_RECALL_AUDIO_SIGNAL
struct	AgsRecallAudioSignal
struct	AgsRecallAudioSignalClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxAnalyseAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxBufferAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxNotationAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxEnvelopeAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxEq10AudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxHighPassAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxLadspaAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxLfoAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxLowPassAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxPatternAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxPeakAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxPlaybackAudioSignal
        &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseAudioSignal
        &#x2570;&#x2500;&#x2500; AgsFxVolumeAudioSignal
```

Implemented Interfaces

AgsRecallAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/ags_recall_audio_signal.h>
```

Description

AgsRecallAudioSignal acts as dynamic audio signal recall. It does audio processing for audio signal context.

Functions

ags_recall_audio_signal_new ()

```
AgsRecallAudioSignal~*
ags_recall_audio_signal_new ();
```

Creates an AgsRecallAudioSignal.

Returns

a new AgsRecallAudioSignal.

Since: 3.0.0

AGS_IS_RECALL_AUDIO_SIGNAL()

```
#define AGS_IS_RECALL_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RECALL_AUDIO_SIGNAL))
```

AGS_IS_RECALL_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_RECALL_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RECALL_AUDIO_SIGNAL))
```

AGS_RECALL_AUDIO_SIGNAL()

```
#define AGS_RECALL_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RECALL_AUDIO_SIGNAL, AgsRecallAudioSignal))
```

AGS_RECALL_AUDIO_SIGNAL_CLASS()

```
#define AGS_RECALL_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RECALL_AUDIO_SIGNAL, AgsRecallAudioSignalClass))
```

AGS_RECALL_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_RECALL_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_RECALL_AUDIO_SIGNAL, AgsRecallAudioSignalClass))
```

ags_recall_audio_signal_get_type ()

```
GType
ags_recall_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_RECALL_AUDIO_SIGNAL

```
#define AGS_TYPE_RECALL_AUDIO_SIGNAL                (ags_recall_audio_signal_get_type())
```

struct AgsRecallAudioSignal

```
struct AgsRecallAudioSignal;
```

struct AgsRecallAudioSignalClass

```
struct AgsRecallAudioSignalClass {
  AgsRecallClass recall;
};
```

Property Details

The "destination" property

```
  "destination"              AgsAudioSignal~*
```

The destination audio signal

Owner: AgsRecallAudioSignal

Flags: Read / Write

Since: 3.0.0

The "source" property

```
  "source"                   AgsAudioSignal~*
```

The source audio signal

Owner: AgsRecallAudioSignal

Flags: Read / Write

Since: 3.0.0

11.15 AgsGenericRecallChannelRun

AgsGenericRecallChannelRun — generic channel context of recall

Stability Level

Stable, unless otherwise indicated

Functions

AgsGenericRecallChannelRun *	ags_generic_recall_channel_run_new ()
#define	AGS_GENERIC_RECALL_CHANNEL_RUN()
#define	AGS_GENERIC_RECALL_CHANNEL_RUN_CLASS()
#define	AGS_GENERIC_RECALL_CHANNEL_RUN_GET_CLASS()
#define	AGS_IS_GENERIC_RECALL_CHANNEL_RUN()
#define	AGS_IS_GENERIC_RECALL_CHANNEL_RUN_CLASS()
GType	ags_generic_recall_channel_run_get_type ()

Properties

GType *	generic-recall-recycling-child-type	Read / Write

Types and Values

#define	AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN
struct	AgsGenericRecallChannelRun
struct	AgsGenericRecallChannelRunClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsGenericRecallChannelRun
```

Implemented Interfaces

AgsGenericRecallChannelRun implements AgsConnectable.

Includes

```
#include <ags/audio/ags_generic_recall_channel_run.h>
```

Description

AgsGenericRecallChannelRun acts as generic channel recall.

Functions

ags_generic_recall_channel_run_new ()

```
AgsGenericRecallChannelRun~*
ags_generic_recall_channel_run_new (AgsChannel *source,
                                    GType child_type,
                                    GType generic_recall_recycling_child_type);
```

Creates an AgsGenericRecallChannelRun.

Parameters

source	the source AgsChannel	
child_type	child type	
generic_recall_recycling_child_type	recycling child type	

Returns

a new AgsGenericRecallChannelRun.

Since: 3.0.0

AGS_GENERIC_RECALL_CHANNEL_RUN()

```
#define AGS_GENERIC_RECALL_CHANNEL_RUN(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN, AgsGenericRecallChannelRun))
```

AGS_GENERIC_RECALL_CHANNEL_RUN_CLASS()

```
#define AGS_GENERIC_RECALL_CHANNEL_RUN_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class) ↩
    , AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN, AgsGenericRecallChannelRunClass))
```

AGS_GENERIC_RECALL_CHANNEL_RUN_GET_CLASS()

```
#define AGS_GENERIC_RECALL_CHANNEL_RUN_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj ↩
    ), AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN, AgsGenericRecallChannelRunClass))
```

AGS_IS_GENERIC_RECALL_CHANNEL_RUN()

```
#define AGS_IS_GENERIC_RECALL_CHANNEL_RUN(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN))
```

AGS_IS_GENERIC_RECALL_CHANNEL_RUN_CLASS()

```
#define AGS_IS_GENERIC_RECALL_CHANNEL_RUN_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN))
```

ags_generic_recall_channel_run_get_type ()

```
GType
ags_generic_recall_channel_run_get_type
                                ();
```

Types and Values

AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN

```
#define AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN                ( ↩
    ags_generic_recall_channel_run_get_type())
```

struct AgsGenericRecallChannelRun

```
struct AgsGenericRecallChannelRun;
```

struct AgsGenericRecallChannelRunClass

```
struct AgsGenericRecallChannelRunClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

Property Details

The "generic-recall-recycling-child-type" property

```
  "generic-recall-recycling-child-type" GType~*
```

The type of child AgsRecall.

Owner: AgsGenericRecallChannelRun

Flags: Read / Write

Allowed values: AgsRecall

Since: 3.0.0

11.16 AgsGenericRecallRecycling

AgsGenericRecallRecycling — generic recycling context of recall

Stability Level

Stable, unless otherwise indicated

Functions

AgsGenericRecallRecycling *	ags_generic_recall_recycling_new ()
#define	AGS_GENERIC_RECALL_RECYCLING()
#define	AGS_GENERIC_RECALL_RECYCLING_CLASS()
#define	AGS_GENERIC_RECALL_RECYCLING_GET_CLASS()
#define	AGS_IS_GENERIC_RECALL_RECYCLING()
#define	AGS_IS_GENERIC_RECALL_RECYCLING_CLASS()
GType	ags_generic_recall_recycling_get_type ()

Types and Values

#define	AGS_TYPE_GENERIC_RECALL_RECYCLING
struct	AgsGenericRecallRecycling
struct	AgsGenericRecallRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsGenericRecallRecycling
```

Implemented Interfaces

AgsGenericRecallRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/ags_generic_recall_recycling.h>
```

Description

AgsGenericRecallRecycling acts as generic recycling recall.

Functions

ags_generic_recall_recycling_new ()

```
AgsGenericRecallRecycling~*
ags_generic_recall_recycling_new (AgsRecycling *recycling,
                                  GType child_type);
```

Creates an AgsGenericRecallRecycling.

Parameters

recycling	the source AgsRecycling	
child_type	child type	

Returns

a new AgsGenericRecallRecycling.

Since: 3.0.0

AGS_GENERIC_RECALL_RECYCLING()

```
#define AGS_GENERIC_RECALL_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_GENERIC_RECALL_RECYCLING, AgsGenericRecallRecycling))
```

AGS_GENERIC_RECALL_RECYCLING_CLASS()

```
#define AGS_GENERIC_RECALL_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
    AGS_TYPE_GENERIC_RECALL_RECYCLING, AgsGenericRecallRecyclingClass))
```

AGS_GENERIC_RECALL_RECYCLING_GET_CLASS()

```
#define AGS_GENERIC_RECALL_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_GENERIC_RECALL_RECYCLING, AgsGenericRecallRecyclingClass))
```

AGS_IS_GENERIC_RECALL_RECYCLING()

```
#define AGS_IS_GENERIC_RECALL_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_GENERIC_RECALL_RECYCLING))
```

AGS_IS_GENERIC_RECALL_RECYCLING_CLASS()

```
#define AGS_IS_GENERIC_RECALL_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_GENERIC_RECALL_RECYCLING))
```

ags_generic_recall_recycling_get_type ()

```
GType
ags_generic_recall_recycling_get_type ();
```

Types and Values

AGS_TYPE_GENERIC_RECALL_RECYCLING

```
#define AGS_TYPE_GENERIC_RECALL_RECYCLING                ( ↩
    ags_generic_recall_recycling_get_type())
```

struct AgsGenericRecallRecycling

```
struct AgsGenericRecallRecycling;
```

struct AgsGenericRecallRecyclingClass

```
struct AgsGenericRecallRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

Chapter 12

Audio thread - multi-threaded audio

Compute audio tree in parallel.

12.1 AgsAudioLoop

AgsAudioLoop — audio loop

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_audio_loop_test_flags ()
void	ags_audio_loop_set_flags ()
void	ags_audio_loop_unset_flags ()
void	ags_audio_loop_add_audio ()
void	ags_audio_loop_remove_audio ()
void	ags_audio_loop_add_channel ()
void	ags_audio_loop_remove_channel ()
gboolean	ags_audio_loop_get_do_fx_staging ()
void	ags_audio_loop_set_do_fx_staging ()
guint *	ags_audio_loop_get_staging_program ()
void	ags_audio_loop_set_staging_program ()
AgsAudioLoop *	ags_audio_loop_new ()
#define	AGS_AUDIO_LOOP()
#define	AGS_AUDIO_LOOP_CLASS()
#define	AGS_AUDIO_LOOP_GET_CLASS()
#define	AGS_IS_AUDIO_LOOP()
#define	AGS_IS_AUDIO_LOOP_CLASS()
GType	ags_audio_loop_get_type ()

Properties

gpointer	play-audio	Write
gpointer	play-channel	Write

Types and Values

#define	AGS_AUDIO_LOOP_DEFAULT_JIFFIE
enum	AgsAudioLoopFlags
#define	AGS_TYPE_AUDIO_LOOP
struct	AgsAudioLoop
struct	AgsAudioLoopClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsThread
        &#x2570;&#x2500;&#x2500; AgsAudioLoop
```

Implemented Interfaces

AgsAudioLoop implements AgsConnectable and AgsMainLoop.

Includes

```
#include <ags/audio/thread/ags_audio_loop.h>
```

Description

The AgsAudioLoop is suitable as AgsMainLoop and does audio processing.

Functions

ags_audio_loop_test_flags ()

```
gboolean
ags_audio_loop_test_flags (AgsAudioLoop *audio_loop,
                           guint flags);
```

Test *flags* to be set on *audio_loop* .

Parameters

audio_loop	the AgsAudioLoop	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_loop_set_flags ()

```
void
ags_audio_loop_set_flags (AgsAudioLoop *audio_loop,
                          guint flags);
```

Set flags.

Parameters

audio_loop	the AgsAudioLoop	
flags	the flags	

Since: 3.0.0

ags_audio_loop_unset_flags ()

```
void
ags_audio_loop_unset_flags (AgsAudioLoop *audio_loop,
                            guint flags);
```

Unset flags.

Parameters

audio_loop	the AgsAudioLoop	
flags	the flags	

Since: 3.0.0

ags_audio_loop_add_audio ()

```
void
ags_audio_loop_add_audio (AgsAudioLoop *audio_loop,
                          GObject *audio);
```

Add audio for playback.

Parameters

audio_loop	the AgsAudioLoop	
audio	an AgsAudio	

Since: 3.0.0

ags_audio_loop_remove_audio ()

```
void
ags_audio_loop_remove_audio (AgsAudioLoop *audio_loop,
                             GObject *audio);
```

Remove audio of playback.

Parameters

audio_loop	the AgsAudioLoop	
audio	an AgsAudio	

Since: 3.0.0

ags_audio_loop_add_channel ()

```
void
ags_audio_loop_add_channel (AgsAudioLoop *audio_loop,
                            GObject *channel);
```

Add channel for playback.

Parameters

audio_loop	the AgsAudioLoop	
channel	an AgsChannel	

Since: 3.0.0

ags_audio_loop_remove_channel ()

```
void
ags_audio_loop_remove_channel (AgsAudioLoop *audio_loop,
                               GObject *channel);
```

Remove channel of playback.

Parameters

audio_loop	the AgsAudioLoop	
channel	an AgsChannel	

Since: 3.0.0

ags_audio_loop_get_do_fx_staging ()

```
gboolean
ags_audio_loop_get_do_fx_staging (AgsAudioLoop *audio_loop);
```

Get do fx staging.

Parameters

audio_loop	the AgsAudioLoop	

Returns

TRUE if set, otherwise FALSE

Since: 3.3.0

ags_audio_loop_set_do_fx_staging ()

```
void
ags_audio_loop_set_do_fx_staging (AgsAudioLoop *audio_loop,
                                  gboolean do_fx_staging);
```

Set do fx staging.

Parameters

audio_loop	the AgsAudioLoop	
do_fx_staging	TRUE if do fx staging, else FALSE	

Since: 3.3.0

ags_audio_loop_get_staging_program ()

```
guint~*
ags_audio_loop_get_staging_program (AgsAudioLoop *audio_loop,
                                    guint *staging_program_count);
```

Get staging program.

Parameters

audio_loop	the AgsAudioLoop	
staging_program_count	the staging program count.	*[out]*

Returns

the staging program.

[transfer full]

Since: 3.3.0

ags_audio_loop_set_staging_program ()

```
void
ags_audio_loop_set_staging_program (AgsAudioLoop *audio_loop,
                                    guint *staging_program,
                                    guint staging_program_count);
```

Set staging program.

Parameters

audio_loop	the AgsAudioLoop	
staging_program	the staging program.	*[transfer none]*
staging_program_count	the staging program count	

Since: 3.3.0

ags_audio_loop_new ()

```
AgsAudioLoop~*
ags_audio_loop_new ();
```

Create a new AgsAudioLoop.

Returns

the new AgsAudioLoop

Since: 3.0.0

AGS_AUDIO_LOOP()

```
#define AGS_AUDIO_LOOP(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_LOOP, AgsAudioLoop))
```

AGS_AUDIO_LOOP_CLASS()

```
#define AGS_AUDIO_LOOP_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_LOOP, AgsAudioLoopClass))
```

AGS_AUDIO_LOOP_GET_CLASS()

```
#define AGS_AUDIO_LOOP_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_LOOP, AgsAudioLoopClass))
```

AGS_IS_AUDIO_LOOP()

```
#define AGS_IS_AUDIO_LOOP(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_LOOP))
```

AGS_IS_AUDIO_LOOP_CLASS()

```
#define AGS_IS_AUDIO_LOOP_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_LOOP))
```

ags_audio_loop_get_type ()

```
GType
ags_audio_loop_get_type ();
```

Types and Values

AGS_AUDIO_LOOP_DEFAULT_JIFFIE

```
#define AGS_AUDIO_LOOP_DEFAULT_JIFFIE (ceil(AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) + AGS_SOUNDCARD_DEFAULT_OVERCLOCK)
```

enum AgsAudioLoopFlags

Enum values to control the behavior or indicate internal state of AgsAudioLoop by enable/disable as flags.

Members

AGS_AUDIO_LOOP_PLAY_CHANNEL	play chan- nel
AGS_AUDIO_LOOP_PLAYING_CHANNEL	playing channnel
AGS_AUDIO_LOOP_PLAY_CHANNEL_TERMINATING	play channe ter- mi- nat- ing
AGS_AUDIO_LOOP_PLAY_AUDIO	play au- dio
AGS_AUDIO_LOOP_PLAYING_AUDIO	playing au- dio
AGS_AUDIO_LOOP_PLAY_AUDIO_TERMINATING	play au- dio ter- mi- nat- ing

AGS_TYPE_AUDIO_LOOP

```
#define AGS_TYPE_AUDIO_LOOP                (ags_audio_loop_get_type())
```

struct AgsAudioLoop

```
struct AgsAudioLoop;
```

struct AgsAudioLoopClass

```
struct AgsAudioLoopClass {
  AgsThreadClass thread;
};
```

Property Details

The "play-audio" property

```
  "play-audio"                gpointer
```

An AgsAudio to add for playback.

[transfer full]

Owner: AgsAudioLoop

Flags: Write

Since: 3.0.0

The "play-channel" property

```
  "play-channel"              gpointer
```

An AgsChannel to add for playback.

[transfer full]

Owner: AgsAudioLoop

Flags: Write

Since: 3.0.0

12.2 AgsAudioThread

AgsAudioThread — audio thread

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_audio_thread_test_status_flags ()
void	ags_audio_thread_set_status_flags ()
void	ags_audio_thread_unset_status_flags ()
gboolean	ags_audio_thread_get_processing ()
void	ags_audio_thread_set_processing ()
AgsTaskLauncher *	ags_audio_thread_get_task_launcher ()
void	ags_audio_thread_set_sound_scope ()
AgsAudioThreadScopeData *	ags_audio_thread_scope_data_alloc ()

void	ags_audio_thread_scope_data_free ()
gboolean	ags_audio_thread_get_do_fx_staging ()
void	ags_audio_thread_set_do_fx_staging ()
guint *	ags_audio_thread_get_staging_program ()
void	ags_audio_thread_set_staging_program ()
AgsAudioThread *	ags_audio_thread_new ()
#define	AGS_AUDIO_THREAD()
#define	AGS_AUDIO_THREAD_CLASS()
#define	AGS_AUDIO_THREAD_GET_CLASS()
#define	AGS_IS_AUDIO_THREAD()
#define	AGS_IS_AUDIO_THREAD_CLASS()
GType	ags_audio_thread_get_type ()

Properties

AgsAudio *	audio	Read / Write
GObject *	default-output-soundcard	Read / Write
gboolean	processing	Read / Write
AgsTaskLauncher *	task-launcher	Read

Types and Values

#define	AGS_AUDIO_THREAD_DEFAULT_JIFFIE
enum	AgsAudioThreadFlags
struct	AgsAudioThreadScopeData
#define	AGS_TYPE_AUDIO_THREAD
struct	AgsAudioThread
struct	AgsAudioThreadClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsThread
    &#x2570;&#x2500;&#x2500; AgsAudioThread
```

Implemented Interfaces

AgsAudioThread implements AgsConnectable.

Includes

```
#include <ags/audio/thread/ags_audio_thread.h>
```

Description

The AgsAudioThread acts as audio output thread to soundcard.

Functions

ags_audio_thread_test_status_flags ()

```
gboolean
ags_audio_thread_test_status_flags (AgsAudioThread *audio_thread,
                                    guint status_flags);
```

Test *status_flags* of *audio_thread* .

Parameters

audio_thread	the AgsAudioThread	
status_flags	status flags	

Returns

TRUE if status flags set, otherwise FALSE

Since: 3.0.0

ags_audio_thread_set_status_flags ()

```
void
ags_audio_thread_set_status_flags (AgsAudioThread *audio_thread,
                                   guint status_flags);
```

Set status flags.

Parameters

audio_thread	the AgsAudioThread	
status_flags	status flags	

Since: 3.0.0

ags_audio_thread_unset_status_flags ()

```
void
ags_audio_thread_unset_status_flags (AgsAudioThread *audio_thread,
                                     guint status_flags);
```

Unset status flags.

Parameters

audio_thread	the AgsAudioThread	
status_flags	status flags	

Since: 3.0.0

ags_audio_thread_get_processing ()

```
gboolean
ags_audio_thread_get_processing (AgsAudioThread *audio_thread);
```

Get processing.

Parameters

audio_thread	the AgsAudioThread

Returns

TRUE if do processing, otherwise FALSE

Since: 3.11.0

ags_audio_thread_set_processing ()

```
void
ags_audio_thread_set_processing (AgsAudioThread *audio_thread,
                                 gboolean processing);
```

Set processing.

Parameters

audio_thread	the AgsAudioThread
processing	TRUE enables and FALSE disables processing

Since: 3.11.0

ags_audio_thread_get_task_launcher ()

```
AgsTaskLauncher~*
ags_audio_thread_get_task_launcher (AgsAudioThread *audio_thread);
```

Get task launcher.

Parameters

audio_thread	the AgsAudioThread

Returns

the AgsTaskLauncher.

[transfer full]

Since: 3.11.0

ags_audio_thread_set_sound_scope ()

```
void
ags_audio_thread_set_sound_scope (AgsAudioThread *audio_thread,
                                  gint sound_scope);
```

Set sound scope.

Parameters

audio_thread	the AgsAudioThread	
sound_scope	the sound scope	

Since: 3.0.0

ags_audio_thread_scope_data_alloc ()

```
AgsAudioThreadScopeData~*
ags_audio_thread_scope_data_alloc ();
```

Allocate the AgsAudioThreadScopeData.

Returns

the newly allocated struct.

[transfer full]

Since: 3.3.0

ags_audio_thread_scope_data_free ()

```
void
ags_audio_thread_scope_data_free (AgsAudioThreadScopeData *scope_data);
```

Free *scope_data* .

Parameters

scope_data	the AgsAudioThreadScopeData.	*[transfer full]*

Since: 3.3.0

ags_audio_thread_get_do_fx_staging ()

```
gboolean
ags_audio_thread_get_do_fx_staging (AgsAudioThread *audio_thread);
```

Get do fx staging.

Parameters

audio_thread	the AgsAudioThread	

Returns

TRUE if set, otherwise FALSE

Since: 3.3.0

ags_audio_thread_set_do_fx_staging ()

```
void
ags_audio_thread_set_do_fx_staging (AgsAudioThread *audio_thread,
                                    gboolean do_fx_staging);
```

Set do fx staging.

Parameters

audio_thread	the AgsAudioThread	
do_fx_staging	TRUE if do fx staging, else FALSE	

Since: 3.3.0

ags_audio_thread_get_staging_program ()

```
guint~*
ags_audio_thread_get_staging_program (AgsAudioThread *audio_thread,
                                      guint *staging_program_count);
```

Get staging program.

Parameters

audio_thread	the AgsAudioThread	
staging_program_count	the staging program count.	*[out]*

Returns

the staging program.

[transfer full]

Since: 3.3.0

ags_audio_thread_set_staging_program ()

```
void
ags_audio_thread_set_staging_program (AgsAudioThread *audio_thread,
                                      guint *staging_program,
                                      guint staging_program_count);
```

Set staging program.

Parameters

audio_thread	the AgsAudioThread	
staging_program	the staging program.	*[transfer none]*
staging_program_count	the staging program count	

Since: 3.3.0

ags_audio_thread_new ()

```
AgsAudioThread~*
ags_audio_thread_new (GObject *default_output_soundcard,
                      GObject *audio);
```

Create a new instance of AgsAudioThread.

Parameters

default_output_soundcard	the GObject	
audio	the AgsAudio	

Returns

the new AgsAudioThread

Since: 3.0.0

AGS_AUDIO_THREAD()

```
#define AGS_AUDIO_THREAD(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_THREAD, AgsAudioThread))
```

AGS_AUDIO_THREAD_CLASS()

```
#define AGS_AUDIO_THREAD_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_THREAD, AgsAudioThreadClass))
```

AGS_AUDIO_THREAD_GET_CLASS()

```
#define AGS_AUDIO_THREAD_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_THREAD, AgsAudioThreadClass))
```

AGS_IS_AUDIO_THREAD()

```
#define AGS_IS_AUDIO_THREAD(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_THREAD))
```

AGS_IS_AUDIO_THREAD_CLASS()

```
#define AGS_IS_AUDIO_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_THREAD))
```

ags_audio_thread_get_type ()

```
GType
ags_audio_thread_get_type ();
```

Types and Values

AGS_AUDIO_THREAD_DEFAULT_JIFFIE

```
#define AGS_AUDIO_THREAD_DEFAULT_JIFFIE (ceil(AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) + AGS_SOUNDCARD_DEFAULT_OVERCLOCK)
```

enum AgsAudioThreadFlags

Enum values to control the behavior or indicate internal state of AgsAudioThread by enable/disable as flags.

Members

AGS_AUDIO_THREAD_STATUS_DONE	sync done par- ent thread, ini- tial wait dur- ing AgsThread::run()
AGS_AUDIO_THREAD_STATUS_WAIT	sync wait par- ent thread, ini- tial wait dur- ing AgsThread::run()

AGS_AUDIO_THREAD_STATUS_DONE_SYNC	sync done parent thread, signal completed during AgsThread::run()
AGS_AUDIO_THREAD_STATUS_WAIT_SYNC	sync wait parent thread, signal completed during AgsThread::run()

struct AgsAudioThreadScopeData

```
struct AgsAudioThreadScopeData {
  volatile gboolean fx_done;
  volatile guint fx_wait;

  GMutex fx_mutex;
  GCond fx_cond;
};
```

AGS_TYPE_AUDIO_THREAD

```
#define AGS_TYPE_AUDIO_THREAD                (ags_audio_thread_get_type())
```

struct AgsAudioThread

```
struct AgsAudioThread;
```

struct AgsAudioThreadClass

```
struct AgsAudioThreadClass {
  AgsThreadClass thread;
};
```

Property Details

The "audio" property

```
"audio"                    AgsAudio~*
```

The assigned AgsAudio.

Owner: AgsAudioThread

Flags: Read / Write

Since: 3.0.0

The "default-output-soundcard" property

```
"default-output-soundcard" GObject~*
```

The assigned default soundcard.

Owner: AgsAudioThread

Flags: Read / Write

Since: 3.0.0

The "processing" property

```
"processing"               gboolean
```

The processing state.

Owner: AgsAudioThread

Flags: Read / Write

Default value: FALSE

Since: 3.11.0

The "task-launcher" property

```
"task-launcher"            AgsTaskLauncher~*
```

The assigned AgsTaskLauncher.

Owner: AgsAudioThread

Flags: Read

Since: 3.11.0

12.3 AgsChannelThread

AgsChannelThread — channel thread

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_channel_thread_test_status_flags ()
void	ags_channel_thread_set_status_flags ()
void	ags_channel_thread_unset_status_flags ()
gboolean	ags_channel_thread_get_processing ()
void	ags_channel_thread_set_processing ()
AgsTaskLauncher *	ags_channel_thread_get_task_launcher ()
void	ags_channel_thread_set_sound_scope ()
gboolean	ags_channel_thread_get_do_fx_staging ()
void	ags_channel_thread_set_do_fx_staging ()
guint *	ags_channel_thread_get_staging_program ()
void	ags_channel_thread_set_staging_program ()
AgsChannelThread *	ags_channel_thread_new ()
#define	AGS_CHANNEL_THREAD()
#define	AGS_CHANNEL_THREAD_CLASS()
#define	AGS_CHANNEL_THREAD_GET_CLASS()
#define	AGS_IS_CHANNEL_THREAD()
#define	AGS_IS_CHANNEL_THREAD_CLASS()
GType	ags_channel_thread_get_type ()

Properties

AgsChannel *	channel	Read / Write
GObject *	default-output-soundcard	Read / Write
gboolean	processing	Read / Write
AgsTaskLauncher *	task-launcher	Read

Types and Values

#define	AGS_CHANNEL_THREAD_DEFAULT_JIFFIE
enum	AgsChannelThreadFlags
#define	AGS_TYPE_CHANNEL_THREAD
struct	AgsChannelThread
struct	AgsChannelThreadClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsThread
        &#x2570;&#x2500;&#x2500; AgsChannelThread
```

Implemented Interfaces

AgsChannelThread implements AgsConnectable.

Includes

```
#include <ags/audio/thread/ags_channel_thread.h>
```

Description

The AgsChannelThread acts as channel output thread to soundcard.

Functions

ags_channel_thread_test_status_flags ()

```
gboolean
ags_channel_thread_test_status_flags (AgsChannelThread *channel_thread,
                                      guint status_flags);
```

Test *status_flags* of *channel_thread*.

Parameters

channel_thread	the AgsChannelThread	
status_flags	status flags	

Returns

TRUE if status flags set, otherwise FALSE

Since: 3.0.0

ags_channel_thread_set_status_flags ()

```
void
ags_channel_thread_set_status_flags (AgsChannelThread *channel_thread,
                                     guint status_flags);
```

Set status flags.

Parameters

channel_thread	the AgsChannelThread	
status_flags	status flags	

Since: 3.0.0

ags_channel_thread_unset_status_flags ()

```
void
ags_channel_thread_unset_status_flags (AgsChannelThread *channel_thread,
                                       guint status_flags);
```

Unset status flags.

Parameters

channel_thread	the AgsChannelThread	
status_flags	status flags	

Since: 3.0.0

ags_channel_thread_get_processing ()

```
gboolean
ags_channel_thread_get_processing (AgsChannelThread *channel_thread);
```

Get processing.

Parameters

channel_thread	the AgsChannelThread	

Returns

TRUE if do processing, otherwise FALSE

Since: 3.11.0

ags_channel_thread_set_processing ()

```
void
ags_channel_thread_set_processing (AgsChannelThread *channel_thread,
                                   gboolean processing);
```

Set processing.

Parameters

channel_thread	the AgsChannelThread	
processing	TRUE enables and FALSE disables processing	

Since: 3.11.0

ags_channel_thread_get_task_launcher ()

```
AgsTaskLauncher~*
ags_channel_thread_get_task_launcher (AgsChannelThread *channel_thread);
```

Get task launcher.

Parameters

channel_thread	the AgsChannelThread	

Returns

the AgsTaskLauncher.

[transfer full]

Since: 3.11.0

ags_channel_thread_set_sound_scope ()

```
void
ags_channel_thread_set_sound_scope (AgsChannelThread *channel_thread,
                                    gint sound_scope);
```

Set sound scope.

Parameters

channel_thread	the AgsChannelThread	
sound_scope	the sound scope	

Since: 3.0.0

ags_channel_thread_get_do_fx_staging ()

```
gboolean
ags_channel_thread_get_do_fx_staging (AgsChannelThread *channel_thread);
```

Get do fx staging.

Parameters

channel_thread	the AgsChannelThread	

Returns

TRUE if set, otherwise FALSE

Since: 3.3.0

ags_channel_thread_set_do_fx_staging ()

```
void
ags_channel_thread_set_do_fx_staging (AgsChannelThread *channel_thread,
                                      gboolean do_fx_staging);
```

Set do fx staging.

Parameters

channel_thread	the AgsChannelThread	
do_fx_staging	TRUE if do fx staging, else FALSe	

Since: 3.3.0

ags_channel_thread_get_staging_program ()

```
guint~*
ags_channel_thread_get_staging_program
                                (AgsChannelThread *channel_thread,
                                 guint *staging_program_count);
```

Get staging program.

Parameters

channel_thread	the AgsChannelThread	
staging_program_count	the staging program count.	*[out]*

Returns

the staging program.

[transfer full]

Since: 3.3.0

ags_channel_thread_set_staging_program ()

```
void
ags_channel_thread_set_staging_program
                                (AgsChannelThread *channel_thread,
                                 guint *staging_program,
                                 guint staging_program_count);
```

Set staging program.

Parameters

channel_thread	the AgsChannelThread	
staging_program	the staging program.	*[transfer none]*
staging_program_count	the staging program count	

Since: 3.3.0

ags_channel_thread_new ()

```
AgsChannelThread~*
ags_channel_thread_new (GObject *default_output_soundcard,
                        GObject *channel);
```

Create a new AgsChannelThread.

Parameters

default_output_soundcard	the GObject	
channel	the AgsChannel	

Returns

the new AgsChannelThread

Since: 3.0.0

AGS_CHANNEL_THREAD()

```
#define AGS_CHANNEL_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CHANNEL_THREAD, AgsChannelThread))
```

AGS_CHANNEL_THREAD_CLASS()

```
#define AGS_CHANNEL_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CHANNEL_THREAD, AgsChannelThreadClass))
```

AGS_CHANNEL_THREAD_GET_CLASS()

```
#define AGS_CHANNEL_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CHANNEL_THREAD, AgsChannelThreadClass))
```

AGS_IS_CHANNEL_THREAD()

```
#define AGS_IS_CHANNEL_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CHANNEL_THREAD))
```

AGS_IS_CHANNEL_THREAD_CLASS()

```
#define AGS_IS_CHANNEL_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CHANNEL_THREAD))
```

ags_channel_thread_get_type ()

```
GType
ags_channel_thread_get_type ();
```

Types and Values

AGS_CHANNEL_THREAD_DEFAULT_JIFFIE

```
#define AGS_CHANNEL_THREAD_DEFAULT_JIFFIE (ceil(AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) + AGS_SOUNDCARD_DEFAULT_OVERCLOCK)
```

enum AgsChannelThreadFlags

Enum values to control the behavior or indicate internal state of AgsChannelThread by enable/disable as flags.

Members

AGS_CHANNEL_THREAD_STATUS_DONE	sync done par- ent thread, ini- tial wait dur- ing AgsThread::run()
AGS_CHANNEL_THREAD_STATUS_WAIT	sync wait par- ent thread, ini- tial wait dur- ing AgsThread::run()
AGS_CHANNEL_THREAD_STATUS_DONE_SYNC	sync done par- ent thread, sig- nal com- pleted dur- ing AgsThread::run()
AGS_CHANNEL_THREAD_STATUS_WAIT_SYNC	sync wait par- ent thread, sig- nal com- pleted dur- ing AgsThread::run()

AGS_TYPE_CHANNEL_THREAD

```
#define AGS_TYPE_CHANNEL_THREAD                (ags_channel_thread_get_type())
```

struct AgsChannelThread

```
struct AgsChannelThread;
```

struct AgsChannelThreadClass

```
struct AgsChannelThreadClass {
  AgsThreadClass thread;
};
```

Property Details

The "channel" property

```
  "channel"                  AgsChannel~*
```

The assigned AgsChannel.

Owner: AgsChannelThread

Flags: Read / Write

Since: 3.0.0

The "default-output-soundcard" property

```
  "default-output-soundcard" GObject~*
```

The assigned default soundcard.

Owner: AgsChannelThread

Flags: Read / Write

Since: 3.0.0

The "processing" property

```
  "processing"               gboolean
```

The processing state.

Owner: AgsChannelThread

Flags: Read / Write

Default value: FALSE

Since: 3.11.0

The "task-launcher" property

```
"task-launcher"            AgsTaskLauncher~*
```

The assigned AgsTaskLauncher.

Owner: AgsChannelThread

Flags: Read

Since: 3.11.0

12.4 AgsExportThread

AgsExportThread — export thread

Stability Level

Stable, unless otherwise indicated

Functions

AgsExportThread *	ags_export_thread_find_soundcard ()
AgsExportThread *	ags_export_thread_new ()
#define	AGS_EXPORT_THREAD()
#define	AGS_EXPORT_THREAD_CLASS()
#define	AGS_EXPORT_THREAD_GET_CLASS()
#define	AGS_IS_EXPORT_THREAD()
#define	AGS_IS_EXPORT_THREAD_CLASS()
GType	ags_export_thread_get_type ()

Properties

AgsAudioFile *	audio-file	Read / Write
GObject *	soundcard	Read / Write
guint	tic	Read / Write

Types and Values

#define	AGS_EXPORT_THREAD_DEFAULT_JIFFIE
enum	AgsExportThreadFlags
#define	AGS_TYPE_EXPORT_THREAD
struct	AgsExportThread
struct	AgsExportThreadClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsThread
    &#x2570;&#x2500;&#x2500; AgsExportThread
```

Implemented Interfaces

AgsExportThread implements AgsConnectable.

Includes

```
#include <ags/audio/thread/ags_export_thread.h>
```

Description

The AgsExportThread acts as audio output thread to file.

Functions

ags_export_thread_find_soundcard ()

```
AgsExportThread~*
ags_export_thread_find_soundcard (AgsExportThread *export_thread,
                                  GObject *soundcard);
```

Parameters

export_thread	the AgsExportThread	
soundcard	the AgsSoundcard to find	

Returns

the matching AgsExportThread, if not found NULL.

[transfer full]

Since: 3.0.0

ags_export_thread_new ()

```
AgsExportThread~*
ags_export_thread_new (GObject *soundcard,
                       AgsAudioFile *audio_file);
```

Create a new instance of AgsExportThread.

Parameters

soundcard	the AgsSoundcard	
audio_file	the output file	

Returns

the new AgsExportThread

Since: 3.0.0

AGS_EXPORT_THREAD()

```
#define AGS_EXPORT_THREAD(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_EXPORT_THREAD, AgsExportThread))
```

AGS_EXPORT_THREAD_CLASS()

```
#define AGS_EXPORT_THREAD_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_EXPORT_THREAD, AgsExportThreadClass))
```

AGS_EXPORT_THREAD_GET_CLASS()

```
#define AGS_EXPORT_THREAD_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_EXPORT_THREAD, AgsExportThreadClass))
```

AGS_IS_EXPORT_THREAD()

```
#define AGS_IS_EXPORT_THREAD(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_EXPORT_THREAD))
```

AGS_IS_EXPORT_THREAD_CLASS()

```
#define AGS_IS_EXPORT_THREAD_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_EXPORT_THREAD))
```

ags_export_thread_get_type ()

```
GType
ags_export_thread_get_type ();
```

Types and Values

AGS_EXPORT_THREAD_DEFAULT_JIFFIE

```
#define AGS_EXPORT_THREAD_DEFAULT_JIFFIE (ceil(AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) + AGS_SOUNDCARD_DEFAULT_OVERCLOCK) // same as  ↩
    soundcard thread
```

enum AgsExportThreadFlags

Enum values to control the behavior or indicate internal state of AgsExportThread by enable/disable as flags.

Members

AGS_EXPORT_THREAD_LIVE_PERFORMANCE	do live export
AGS_EXPORT_THREAD_IS_EXPORTING	

AGS_TYPE_EXPORT_THREAD

```
#define AGS_TYPE_EXPORT_THREAD                (ags_export_thread_get_type())
```

struct AgsExportThread

```
struct AgsExportThread;
```

struct AgsExportThreadClass

```
struct AgsExportThreadClass {
  AgsThreadClass thread;
};
```

Property Details

The "audio-file" property

```
  "audio-file"                AgsAudioFile~*
```

The assigned AgsAudioFile.

Owner: AgsExportThread

Flags: Read / Write

Since: 3.0.0

The "soundcard" property

```
  "soundcard"                 GObject~*
```

The assigned AgsSoundcard.

Owner: AgsExportThread

Flags: Read / Write

Since: 3.0.0

The "tic" property

```
  "tic"                       guint
```

The tic.

Owner: AgsExportThread

Flags: Read / Write

Default value: 0

Since: 3.0.0

12.5 AgsSequencerThread

AgsSequencerThread — sequencer thread

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_sequencer_thread_interval_timeout ()
AgsSequencerThread *	ags_sequencer_thread_find_sequencer ()
AgsSequencerThread *	ags_sequencer_thread_new ()
#define	AGS_IS_SEQUENCER_THREAD()
#define	AGS_IS_SEQUENCER_THREAD_CLASS()
#define	AGS_SEQUENCER_THREAD()
#define	AGS_SEQUENCER_THREAD_CLASS()
#define	AGS_SEQUENCER_THREAD_GET_CLASS()
GType	ags_sequencer_thread_get_type ()

Properties

GObject *	sequencer	Write

Signals

void	interval-timeout	Run Last

Types and Values

#define	AGS_SEQUENCER_THREAD_DEFAULT_JIFFIE
#define	AGS_TYPE_SEQUENCER_THREAD
struct	AgsSequencerThread
struct	AgsSequencerThreadClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsThread
    &#x2570;&#x2500;&#x2500; AgsSequencerThread
```

Implemented Interfaces

AgsSequencerThread implements AgsConnectable.

Includes

```
#include <ags/thread/ags_sequencer_thread.h>
```

Description

The AgsSequencerThread acts as midi input thread.

Functions

ags_sequencer_thread_interval_timeout ()

```
void
ags_sequencer_thread_interval_timeout (AgsSequencerThread *sequencer_thread);
```

ags_sequencer_thread_find_sequencer ()

```
AgsSequencerThread~*
ags_sequencer_thread_find_sequencer (AgsSequencerThread *sequencer_thread,
                                     GObject *sequencer);
```

Find *sequencer* as sibling of *sequencer_thread* , if it was found on the returned thread should be called g_object_unref().

Parameters

sequencer_thread	the AgsSequencerThread	
sequencer	the AgsSequencer to find	

Returns

the matching AgsSequencerThread, if not found NULL.

[transfer full]

Since: 3.0.0

ags_sequencer_thread_new ()

```
AgsSequencerThread~*
ags_sequencer_thread_new (GObject *sequencer);
```

Create a new instance of AgsSequencerThread.

Parameters

sequencer	the AgsSequencer	

Returns

the new AgsSequencerThread

Since: 3.0.0

AGS_IS_SEQUENCER_THREAD()

```
#define AGS_IS_SEQUENCER_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SEQUENCER_THREAD))
```

AGS_IS_SEQUENCER_THREAD_CLASS()

```
#define AGS_IS_SEQUENCER_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SEQUENCER_THREAD))
```

AGS_SEQUENCER_THREAD()

```
#define AGS_SEQUENCER_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SEQUENCER_THREAD, AgsSequencerThread))
```

AGS_SEQUENCER_THREAD_CLASS()

```
#define AGS_SEQUENCER_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_SEQUENCER_THREAD, AgsSequencerThreadClass))
```

AGS_SEQUENCER_THREAD_GET_CLASS()

```
#define AGS_SEQUENCER_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_SEQUENCER_THREAD, AgsSequencerThreadClass))
```

ags_sequencer_thread_get_type ()

```
GType
ags_sequencer_thread_get_type ();
```

Types and Values

AGS_SEQUENCER_THREAD_DEFAULT_JIFFIE

```
#define AGS_SEQUENCER_THREAD_DEFAULT_JIFFIE (ceil(AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) + AGS_SOUNDCARD_DEFAULT_OVERCLOCK)
```

AGS_TYPE_SEQUENCER_THREAD

```
#define AGS_TYPE_SEQUENCER_THREAD                 (ags_sequencer_thread_get_type())
```

struct AgsSequencerThread

```
struct AgsSequencerThread;
```

struct AgsSequencerThreadClass

```
struct AgsSequencerThreadClass {
  AgsThreadClass thread;

  void (*interval_timeout)(AgsSequencerThread *sequencer_thread);
};
```

Property Details

The "sequencer" property

```
  "sequencer"                GObject~*
```

The assigned AgsSequencer.

Owner: AgsSequencerThread

Flags: Write

Since: 3.0.0

Signal Details

The "interval-timeout" signal

```
void
user_function (AgsSequencerThread *agssequencerthread,
               gpointer            user_data)
```

Flags: Run Last

12.6 AgsSoundcardThread

AgsSoundcardThread — soundcard thread

Stability Level

Stable, unless otherwise indicated

Functions

AgsSoundcardThread *	ags_soundcard_thread_find_soundcard ()
AgsSoundcardThread *	ags_soundcard_thread_new ()
#define	AGS_IS_SOUNDCARD_THREAD()
#define	AGS_IS_SOUNDCARD_THREAD_CLASS()
#define	AGS_SOUNDCARD_THREAD()
#define	AGS_SOUNDCARD_THREAD_CLASS()
#define	AGS_SOUNDCARD_THREAD_GET_CLASS()
GType	ags_soundcard_thread_get_type ()

Properties

GObject *	soundcard	Read / Write
guint	soundcard-capability	Read / Write

Types and Values

#define	AGS_SOUNDCARD_THREAD_DEFAULT_JIFFIE
#define	AGS_TYPE_SOUNDCARD_THREAD
struct	AgsSoundcardThread
struct	AgsSoundcardThreadClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsThread
        &#x2570;&#x2500;&#x2500; AgsSoundcardThread
```

Implemented Interfaces

AgsSoundcardThread implements AgsConnectable.

Includes

```
#include <ags/audio/thread/ags_soundcard_thread.h>
```

Description

The AgsSoundcardThread acts as audio output thread to soundcard.

Functions

ags_soundcard_thread_find_soundcard ()

```
AgsSoundcardThread~*
ags_soundcard_thread_find_soundcard (AgsSoundcardThread *soundcard_thread,
                                     GObject *soundcard);
```

Find *soundcard* as sibling of *soundcard_thread* , if it was found on the returned thread should be called g_object_unref().

Parameters

soundcard_thread	the AgsSoundcardThread	
soundcard	the AgsSoundcard to find	

Returns

the matching AgsSoundcardThread, if not found NULL.

[transfer full]

Since: 3.0.0

ags_soundcard_thread_new ()

```
AgsSoundcardThread~*
ags_soundcard_thread_new (GObject *soundcard,
                          guint soundcard_capability);
```

Create a new instance of AgsSoundcardThread.

Parameters

soundcard	the AgsSoundcard	
soundcard_capability	see AgsSoundcardCapability	

Returns

the new AgsSoundcardThread

Since: 3.0.0

AGS_IS_SOUNDCARD_THREAD()

```
#define AGS_IS_SOUNDCARD_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SOUNDCARD_THREAD))
```

AGS_IS_SOUNDCARD_THREAD_CLASS()

```
#define AGS_IS_SOUNDCARD_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SOUNDCARD_THREAD))
```

AGS_SOUNDCARD_THREAD()

```
#define AGS_SOUNDCARD_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOUNDCARD_THREAD, AgsSoundcardThread))
```

AGS_SOUNDCARD_THREAD_CLASS()

```
#define AGS_SOUNDCARD_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_SOUNDCARD_THREAD, AgsSoundcardThreadClass))
```

AGS_SOUNDCARD_THREAD_GET_CLASS()

```
#define AGS_SOUNDCARD_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_SOUNDCARD_THREAD, AgsSoundcardThreadClass))
```

ags_soundcard_thread_get_type ()

```
GType
ags_soundcard_thread_get_type ();
```

Types and Values

AGS_SOUNDCARD_THREAD_DEFAULT_JIFFIE

```
#define AGS_SOUNDCARD_THREAD_DEFAULT_JIFFIE (ceil(AGS_SOUNDCARD_DEFAULT_SAMPLERATE / ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) + AGS_SOUNDCARD_DEFAULT_OVERCLOCK)
```

AGS_TYPE_SOUNDCARD_THREAD

```
#define AGS_TYPE_SOUNDCARD_THREAD                (ags_soundcard_thread_get_type())
```

struct AgsSoundcardThread

```
struct AgsSoundcardThread;
```

struct AgsSoundcardThreadClass

```
struct AgsSoundcardThreadClass {
  AgsThreadClass thread;
};
```

Property Details

The "soundcard" property

```
  "soundcard"                GObject~*
```

The assigned AgsSoundcard.

Owner: AgsSoundcardThread

Flags: Read / Write

Since: 3.0.0

The "soundcard-capability" property

```
  "soundcard-capability"     guint
```

The soundcard capability.

Owner: AgsSoundcardThread

Flags: Read / Write

Allowed values: <= G_MAXINT

Default value: 0

Since: 3.0.0

12.7 AgsSF2Loader

AgsSF2Loader — load Soundfont2 asynchronously

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SF2_LOADER_GET_OBJ_MUTEX()
gboolean	ags_sf2_loader_test_flags ()
void	ags_sf2_loader_set_flags ()
void	ags_sf2_loader_unset_flags ()
AgsAudio *	ags_sf2_loader_get_audio ()
void	ags_sf2_loader_set_audio ()
gchar *	ags_sf2_loader_get_filename ()
void	ags_sf2_loader_set_filename ()
gchar *	ags_sf2_loader_get_preset ()
void	ags_sf2_loader_set_preset ()
gchar *	ags_sf2_loader_get_instrument ()
void	ags_sf2_loader_set_instrument ()
AgsAudioContainer *	ags_sf2_loader_get_audio_container ()
void	ags_sf2_loader_set_audio_container ()
void	ags_sf2_loader_start ()
AgsSF2Loader *	ags_sf2_loader_new ()
#define	AGS_IS_SF2_LOADER()
#define	AGS_IS_SF2_LOADER_CLASS()
#define	AGS_SF2_LOADER()
#define	AGS_SF2_LOADER_CLASS()
#define	AGS_SF2_LOADER_GET_CLASS()
GType	ags_sf2_loader_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsAudioContainer *	audio-container	Read / Write
char *	filename	Read / Write
char *	instrument	Read / Write
char *	preset	Read / Write

Types and Values

enum	AgsSF2LoaderFlags
#define	AGS_TYPE_SF2_LOADER
struct	AgsSF2Loader
struct	AgsSF2LoaderClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsSF2Loader
```

Includes

```
#include <ags/plugin/ags_sf2_loader.h>
```

Description

The AgsSF2Loader loads Soundfont2 audio data asynchronously.

Functions

AGS_SF2_LOADER_GET_OBJ_MUTEX()

```
#define AGS_SF2_LOADER_GET_OBJ_MUTEX(obj) (&(((AgsSF2Loader *) obj)->obj_mutex))
```

ags_sf2_loader_test_flags ()

```
gboolean
ags_sf2_loader_test_flags (AgsSF2Loader *sf2_loader,
                           guint flags);
```

Test *flags* to be set on *recall* .

Parameters

sf2_loader	the AgsSF2Loader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sf2_loader_set_flags ()

```
void
ags_sf2_loader_set_flags (AgsSF2Loader *sf2_loader,
                          guint flags);
```

Set flags.

Parameters

sf2_loader	the AgsSF2Loader	
flags	the flags	

Since: 3.0.0

ags_sf2_loader_unset_flags ()

```
void
ags_sf2_loader_unset_flags (AgsSF2Loader *sf2_loader,
                            guint flags);
```

Unset flags.

Parameters

sf2_loader	the AgsSF2Loader	
flags	the flags	

Since: 3.0.0

ags_sf2_loader_get_audio ()

```
AgsAudio~*
ags_sf2_loader_get_audio (AgsSF2Loader *sf2_loader);
```

Get audio of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	

Returns

the assigned AgsAudio.

[transfer full]

Since: 3.2.0

ags_sf2_loader_set_audio ()

```
void
ags_sf2_loader_set_audio (AgsSF2Loader *sf2_loader,
                          AgsAudio *audio);
```

Set audio of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	
audio	the AgsAudio	

Since: 3.2.0

ags_sf2_loader_get_filename ()

```
gchar~*
ags_sf2_loader_get_filename (AgsSF2Loader *sf2_loader);
```

Get filename of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	

Returns

the assigned filename.

[transfer full]

Since: 3.2.0

ags_sf2_loader_set_filename ()

```
void
ags_sf2_loader_set_filename (AgsSF2Loader *sf2_loader,
                             gchar *filename);
```

Set filename of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	
filename	the filename	

Since: 3.2.0

ags_sf2_loader_get_preset ()

```
gchar~*
ags_sf2_loader_get_preset (AgsSF2Loader *sf2_loader);
```

Get preset of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	

Returns

the assigned preset.

[transfer full]

Since: 3.2.0

ags_sf2_loader_set_preset ()

```
void
ags_sf2_loader_set_preset (AgsSF2Loader *sf2_loader,
                           gchar *preset);
```

Set preset of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	
preset	the preset	

Since: 3.2.0

ags_sf2_loader_get_instrument ()

```
gchar~*
ags_sf2_loader_get_instrument (AgsSF2Loader *sf2_loader);
```

Get instrument of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	

Returns

the assigned instrument.

[transfer full]

Since: 3.2.0

ags_sf2_loader_set_instrument ()

```
void
ags_sf2_loader_set_instrument (AgsSF2Loader *sf2_loader,
                               gchar *instrument);
```

Set instrument of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	
instrument	the instrument	

Since: 3.2.0

ags_sf2_loader_get_audio_container ()

```
AgsAudioContainer~*
ags_sf2_loader_get_audio_container (AgsSF2Loader *sf2_loader);
```

Get audio container of *sf2_loader* .

Parameters

sf2_loader	the AgsSF2Loader	

Returns

the assigned AgsAudioContainer.

[transfer full]

Since: 3.2.0

ags_sf2_loader_set_audio_container ()

```
void
ags_sf2_loader_set_audio_container (AgsSF2Loader *sf2_loader,
                                    AgsAudioContainer *audio_container);
```

Set audio container of *sf2_loader*.

Parameters

sf2_loader	the AgsSF2Loader	
audio_container	the AgsAudioContainer	

Since: 3.2.0

ags_sf2_loader_start ()

```
void
ags_sf2_loader_start (AgsSF2Loader *sf2_loader);
```

Start *sf2_loader*.

Parameters

sf2_loader	the AgsSF2Loader	

Since: 3.0.0

ags_sf2_loader_new ()

```
AgsSF2Loader~*
ags_sf2_loader_new (AgsAudio *audio,
                    gchar *filename,
                    gchar *preset,
                    gchar *instrument);
```

Create a new instance of AgsSF2Loader.

Parameters

audio	the AgsAudio	
filename	the filename	
preset	the preset	
instrument	the instrument	

Returns

the new AgsSF2Loader

Since: 3.0.0

AGS_IS_SF2_LOADER()

```
#define AGS_IS_SF2_LOADER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SF2_LOADER))
```

AGS_IS_SF2_LOADER_CLASS()

```
#define AGS_IS_SF2_LOADER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SF2_LOADER))
```

AGS_SF2_LOADER()

```
#define AGS_SF2_LOADER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SF2_LOADER, AgsSF2Loader))
```

AGS_SF2_LOADER_CLASS()

```
#define AGS_SF2_LOADER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SF2_LOADER, AgsSF2LoaderClass))
```

AGS_SF2_LOADER_GET_CLASS()

```
#define AGS_SF2_LOADER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SF2_LOADER, AgsSF2LoaderClass))
```

ags_sf2_loader_get_type ()

```
GType
ags_sf2_loader_get_type ();
```

Types and Values

enum AgsSF2LoaderFlags

Enum values to configure SF2 loader.

Members

AGS_SF2_LOADER_HAS_COMPLETED	has com- pleted
AGS_SF2_LOADER_RUN_APPLY_SYNTH	run ap- ply synth

AGS_TYPE_SF2_LOADER

```
#define AGS_TYPE_SF2_LOADER                (ags_sf2_loader_get_type())
```

struct AgsSF2Loader

```
struct AgsSF2Loader;
```

struct AgsSF2LoaderClass

```
struct AgsSF2LoaderClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned audio.

Owner: AgsSF2Loader

Flags: Read / Write

Since: 3.0.0

The "audio-container" property

```
  "audio-container"          AgsAudioContainer~*
```

The audio container opened.

Owner: AgsSF2Loader

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
"filename"                  char~*
```

The filename to open.

Owner: AgsSF2Loader

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "instrument" property

```
"instrument"                char~*
```

The instrument to open.

Owner: AgsSF2Loader

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "preset" property

```
"preset"                    char~*
```

The preset to open.

Owner: AgsSF2Loader

Flags: Read / Write

Default value: NULL

Since: 3.0.0

12.8 AgsSF2MidiLocaleLoader

AgsSF2MidiLocaleLoader — load Soundfont2 MIDI locale asynchronously

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SF2_MIDI_LOCALE_LOADER_GET_OBJ_MUTEX()
gboolean	ags_sf2_midi_locale_loader_test_flags ()
void	ags_sf2_midi_locale_loader_set_flags ()
void	ags_sf2_midi_locale_loader_unset_flags ()

AgsAudio *	ags_sf2_midi_locale_loader_get_audio ()
void	ags_sf2_midi_locale_loader_set_audio ()
gchar *	ags_sf2_midi_locale_loader_get_filename ()
void	ags_sf2_midi_locale_loader_set_filename ()
gint	ags_sf2_midi_locale_loader_get_bank ()
void	ags_sf2_midi_locale_loader_set_bank ()
gint	ags_sf2_midi_locale_loader_get_program ()
void	ags_sf2_midi_locale_loader_set_program ()
AgsAudioContainer *	ags_sf2_midi_locale_loader_get_audio_container ()
void	ags_sf2_midi_locale_loader_set_audio_container ()
AgsSF2SynthUtil *	ags_sf2_midi_locale_loader_get_template ()
void	ags_sf2_midi_locale_loader_set_template ()
AgsSF2SynthUtil *	ags_sf2_midi_locale_loader_get_synth ()
void	ags_sf2_midi_locale_loader_set_synth ()
void	ags_sf2_midi_locale_loader_start ()
AgsSF2MidiLocaleLoader *	ags_sf2_midi_locale_loader_new ()
#define	AGS_IS_SF2_MIDI_LOCALE_LOADER()
#define	AGS_IS_SF2_MIDI_LOCALE_LOADER_CLASS()
#define	AGS_SF2_MIDI_LOCALE_LOADER()
#define	AGS_SF2_MIDI_LOCALE_LOADER_CLASS()
#define	AGS_SF2_MIDI_LOCALE_LOADER_GET_CLASS()
GType	ags_sf2_midi_locale_loader_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsAudioContainer *	audio-container	Read / Write
int	bank	Read / Write
char *	filename	Read / Write
int	program	Read / Write
gpointer	synth	Read / Write
gpointer	template	Read / Write

Types and Values

enum	AgsSF2MidiLocaleLoaderFlags
#define	AGS_TYPE_SF2_MIDI_LOCALE_LOADER
struct	AgsSF2MidiLocaleLoader
struct	AgsSF2MidiLocaleLoaderClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsSF2MidiLocaleLoader
```

Includes

```
#include <ags/plugin/ags_sf2_midi_locale_loader.h>
```

Description

The AgsSF2MidiLocaleLoader loads Soundfont2 audio data of MIDI locale asynchronously.

Functions

AGS_SF2_MIDI_LOCALE_LOADER_GET_OBJ_MUTEX()

```
#define AGS_SF2_MIDI_LOCALE_LOADER_GET_OBJ_MUTEX(obj) (&(((AgsSF2MidiLocaleLoader *) obj)-> ↩
    obj_mutex))
```

ags_sf2_midi_locale_loader_test_flags ()

```
gboolean
ags_sf2_midi_locale_loader_test_flags (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                       guint flags);
```

Test *flags* to be set on *recall* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.16.0

ags_sf2_midi_locale_loader_set_flags ()

```
void
ags_sf2_midi_locale_loader_set_flags (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                      guint flags);
```

Set flags.

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
flags	the flags	

Since: 3.16.0

ags_sf2_midi_locale_loader_unset_flags ()

```
void
ags_sf2_midi_locale_loader_unset_flags
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                 guint flags);
```

Unset flags.

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
flags	the flags	

Since: 3.16.0

ags_sf2_midi_locale_loader_get_audio ()

```
AgsAudio~*
ags_sf2_midi_locale_loader_get_audio (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get audio of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	

Returns

the assigned AgsAudio.

[transfer full]

Since: 3.16.0

ags_sf2_midi_locale_loader_set_audio ()

```
void
ags_sf2_midi_locale_loader_set_audio (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                      AgsAudio *audio);
```

Set audio of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
audio	the AgsAudio	

Since: 3.16.0

ags_sf2_midi_locale_loader_get_filename ()

```
gchar~*
ags_sf2_midi_locale_loader_get_filename
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get filename of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader

Returns

the assigned filename.

[transfer full]

Since: 3.16.0

ags_sf2_midi_locale_loader_set_filename ()

```
void
ags_sf2_midi_locale_loader_set_filename
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                 gchar *filename);
```

Set filename of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader
filename	the filename

Since: 3.16.0

ags_sf2_midi_locale_loader_get_bank ()

```
gint
ags_sf2_midi_locale_loader_get_bank (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get bank of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader

Returns

the assigned bank

Since: 3.16.0

ags_sf2_midi_locale_loader_set_bank ()

```
void
ags_sf2_midi_locale_loader_set_bank (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                     gint bank);
```

Set bank of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
bank	the bank	

Since: 3.16.0

ags_sf2_midi_locale_loader_get_program ()

```
gint
ags_sf2_midi_locale_loader_get_program
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get program of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	

Returns

the assigned program

Since: 3.16.0

ags_sf2_midi_locale_loader_set_program ()

```
void
ags_sf2_midi_locale_loader_set_program
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                 gint program);
```

Set program of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
program	the program	

Since: 3.16.0

ags_sf2_midi_locale_loader_get_audio_container ()

```
AgsAudioContainer~*
ags_sf2_midi_locale_loader_get_audio_container
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get audio container of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	

Returns

the assigned AgsAudioContainer.

[transfer full]

Since: 3.16.0

ags_sf2_midi_locale_loader_set_audio_container ()

```
void
ags_sf2_midi_locale_loader_set_audio_container
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                 AgsAudioContainer *audio_container);
```

Set audio container of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
audio_container	the AgsAudioContainer	

Since: 3.16.0

ags_sf2_midi_locale_loader_get_template ()

```
AgsSF2SynthUtil~*
ags_sf2_midi_locale_loader_get_template
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get AgsSF2SynthUtil template of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	

Returns

the assigned AgsSF2SynthUtil

Since: 3.16.0

ags_sf2_midi_locale_loader_set_template ()

```
void
ags_sf2_midi_locale_loader_set_template
                                (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                 AgsSF2SynthUtil *template);
```

Set AgsSF2SynthUtil template of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
template	the AgsSF2SynthUtil	

Since: 3.16.0

ags_sf2_midi_locale_loader_get_synth ()

```
AgsSF2SynthUtil~*
ags_sf2_midi_locale_loader_get_synth (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Get AgsSF2SynthUtil synth of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	

Returns

the assigned AgsSF2SynthUtil

Since: 3.16.0

ags_sf2_midi_locale_loader_set_synth ()

```
void
ags_sf2_midi_locale_loader_set_synth (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader,
                                      AgsSF2SynthUtil *synth);
```

Set AgsSF2SynthUtil synth of *sf2_midi_locale_loader* .

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	
synth	the AgsSF2SynthUtil	

Since: 3.16.0

ags_sf2_midi_locale_loader_start ()

```
void
ags_sf2_midi_locale_loader_start (AgsSF2MidiLocaleLoader *sf2_midi_locale_loader);
```

Start *sf2_midi_locale_loader*.

Parameters

sf2_midi_locale_loader	the AgsSF2MidiLocaleLoader	

Since: 3.16.0

ags_sf2_midi_locale_loader_new ()

```
AgsSF2MidiLocaleLoader~*
ags_sf2_midi_locale_loader_new (AgsAudio *audio,
                                gchar *filename,
                                gint bank,
                                gint program);
```

Create a new instance of AgsSF2MidiLocaleLoader.

Parameters

audio	the AgsAudio	
filename	the filename	
bank	the bank	
program	the program	

Returns

the new AgsSF2MidiLocaleLoader

Since: 3.16.0

AGS_IS_SF2_MIDI_LOCALE_LOADER()

```
#define AGS_IS_SF2_MIDI_LOCALE_LOADER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
    AGS_TYPE_SF2_MIDI_LOCALE_LOADER))
```

AGS_IS_SF2_MIDI_LOCALE_LOADER_CLASS()

```
#define AGS_IS_SF2_MIDI_LOCALE_LOADER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
    AGS_TYPE_SF2_MIDI_LOCALE_LOADER))
```

AGS_SF2_MIDI_LOCALE_LOADER()

```
#define AGS_SF2_MIDI_LOCALE_LOADER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_SF2_MIDI_LOCALE_LOADER, AgsSF2MidiLocaleLoader))
```

AGS_SF2_MIDI_LOCALE_LOADER_CLASS()

```
#define AGS_SF2_MIDI_LOCALE_LOADER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SF2_MIDI_LOCALE_LOADER, AgsSF2MidiLocaleLoaderClass))
```

AGS_SF2_MIDI_LOCALE_LOADER_GET_CLASS()

```
#define AGS_SF2_MIDI_LOCALE_LOADER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SF2_MIDI_LOCALE_LOADER, AgsSF2MidiLocaleLoaderClass))
```

ags_sf2_midi_locale_loader_get_type ()

```
GType
ags_sf2_midi_locale_loader_get_type ();
```

Types and Values

enum AgsSF2MidiLocaleLoaderFlags

Enum values to configure Soundfont2 MIDI locale loader.

Members

AGS_SF2_MIDI_LOCALE_LOADER_HAS_COMPLETED	has completed
AGS_SF2_MIDI_LOCALE_LOADER_RUN_APPLY_MIDI_LOCALE	

AGS_TYPE_SF2_MIDI_LOCALE_LOADER

```
#define AGS_TYPE_SF2_MIDI_LOCALE_LOADER                (ags_sf2_midi_locale_loader_get_type ↩
    ())
```

struct AgsSF2MidiLocaleLoader

```
struct AgsSF2MidiLocaleLoader;
```

struct AgsSF2MidiLocaleLoaderClass

```
struct AgsSF2MidiLocaleLoaderClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
"audio"                    AgsAudio~*
```

The assigned audio.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Since: 3.16.0

The "audio-container" property

```
"audio-container"          AgsAudioContainer~*
```

The audio container opened.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Since: 3.16.0

The "bank" property

```
"bank"                     int
```

The bank to open.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Allowed values: [-1,128]

Default value: -1

Since: 3.16.0

The "filename" property

```
"filename"                 char~*
```

The filename to open.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Default value: NULL

Since: 3.16.0

The "program" property

```
"program"                    int
```

The program to open.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Allowed values: [-1,128]

Default value: -1

Since: 3.16.0

The "synth" property

```
"synth"                      gpointer
```

The assigned synth.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Since: 3.16.0

The "template" property

```
"template"                   gpointer
```

The assigned template.

Owner: AgsSF2MidiLocaleLoader

Flags: Read / Write

Since: 3.16.0

12.9 AgsSFZLoader

AgsSFZLoader — load sfz asynchronously

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_LOADER_GET_OBJ_MUTEX()
gboolean	ags_sfz_loader_test_flags ()
void	ags_sfz_loader_set_flags ()
void	ags_sfz_loader_unset_flags ()
AgsAudio *	ags_sfz_loader_get_audio ()
void	ags_sfz_loader_set_audio ()

gchar *	ags_sfz_loader_get_filename ()
void	ags_sfz_loader_set_filename ()
AgsAudioContainer *	ags_sfz_loader_get_audio_container ()
void	ags_sfz_loader_set_audio_container ()
void	ags_sfz_loader_start ()
AgsSFZLoader *	ags_sfz_loader_new ()
#define	AGS_IS_SFZ_LOADER()
#define	AGS_IS_SFZ_LOADER_CLASS()
#define	AGS_SFZ_LOADER()
#define	AGS_SFZ_LOADER_CLASS()
#define	AGS_SFZ_LOADER_GET_CLASS()
GType	ags_sfz_loader_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsAudioContainer *	audio-container	Read / Write
char *	filename	Read / Write

Types and Values

enum	AgsSFZLoaderFlags
#define	AGS_TYPE_SFZ_LOADER
struct	AgsSFZLoader
struct	AgsSFZLoaderClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsSFZLoader
```

Includes

```
#include <ags/plugin/ags_sfz_loader.h>
```

Description

The AgsSFZLoader loads sfz audio data asynchronously.

Functions

AGS_SFZ_LOADER_GET_OBJ_MUTEX()

```
#define AGS_SFZ_LOADER_GET_OBJ_MUTEX(obj) (&(((AgsSFZLoader *) obj)->obj_mutex))
```

ags_sfz_loader_test_flags ()

```
gboolean
ags_sfz_loader_test_flags (AgsSFZLoader *sfz_loader,
                           guint flags);
```

Test *flags* to be set on *recall* .

Parameters

sfz_loader	the AgsSFZLoader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sfz_loader_set_flags ()

```
void
ags_sfz_loader_set_flags (AgsSFZLoader *sfz_loader,
                          guint flags);
```

Set flags.

Parameters

sfz_loader	the AgsSFZLoader	
flags	the flags	

Since: 3.0.0

ags_sfz_loader_unset_flags ()

```
void
ags_sfz_loader_unset_flags (AgsSFZLoader *sfz_loader,
                            guint flags);
```

Unset flags.

Parameters

sfz_loader	the AgsSFZLoader	
flags	the flags	

Since: 3.0.0

ags_sfz_loader_get_audio ()

```
AgsAudio~*
ags_sfz_loader_get_audio (AgsSFZLoader *sfz_loader);
```

Get audio of *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader

Returns

the assigned AgsAudio.

[transfer full]

Since: 3.2.0

ags_sfz_loader_set_audio ()

```
void
ags_sfz_loader_set_audio (AgsSFZLoader *sfz_loader,
                          AgsAudio *audio);
```

Set audio of *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader
audio	the AgsAudio

Since: 3.2.0

ags_sfz_loader_get_filename ()

```
gchar~*
ags_sfz_loader_get_filename (AgsSFZLoader *sfz_loader);
```

Get filename of *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader

Returns

the assigned filename.

[transfer full]

Since: 3.2.0

ags_sfz_loader_set_filename ()

```
void
ags_sfz_loader_set_filename (AgsSFZLoader *sfz_loader,
                             gchar *filename);
```

Set filename of *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader	
filename	the filename	

Since: 3.2.0

ags_sfz_loader_get_audio_container ()

```
AgsAudioContainer~*
ags_sfz_loader_get_audio_container (AgsSFZLoader *sfz_loader);
```

Get audio container of *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader	

Returns

the assigned AgsAudioContainer.

[transfer full]

Since: 3.2.0

ags_sfz_loader_set_audio_container ()

```
void
ags_sfz_loader_set_audio_container (AgsSFZLoader *sfz_loader,
                                    AgsAudioContainer *audio_container);
```

Set audio container of *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader	
audio_container	the AgsAudioContainer	

Since: 3.2.0

ags_sfz_loader_start ()

```
void
ags_sfz_loader_start (AgsSFZLoader *sfz_loader);
```

Start *sfz_loader* .

Parameters

sfz_loader	the AgsSFZLoader	

Since: 3.0.0

ags_sfz_loader_new ()

```
AgsSFZLoader~*
ags_sfz_loader_new (AgsAudio *audio,
                    gchar *filename,
                    gboolean do_replace);
```

Create a new instance of AgsSFZLoader.

Parameters

audio	the AgsAudio	
filename	the filename	
do_replace	if TRUE replace *audio* 's sfz	

Returns

the new AgsSFZLoader

Since: 3.0.0

AGS_IS_SFZ_LOADER()

```
#define AGS_IS_SFZ_LOADER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SFZ_LOADER))
```

AGS_IS_SFZ_LOADER_CLASS()

```
#define AGS_IS_SFZ_LOADER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SFZ_LOADER))
```

AGS_SFZ_LOADER()

```
#define AGS_SFZ_LOADER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_LOADER, AgsSFZLoader))
```

AGS_SFZ_LOADER_CLASS()

```
#define AGS_SFZ_LOADER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_LOADER, AgsSFZLoaderClass))
```

AGS_SFZ_LOADER_GET_CLASS()

```
#define AGS_SFZ_LOADER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SFZ_LOADER, AgsSFZLoaderClass))
```

ags_sfz_loader_get_type ()

```
GType
ags_sfz_loader_get_type ();
```

Types and Values

enum AgsSFZLoaderFlags

Enum values to configure SFZ loader.

Members

AGS_SFZ_LOADER_DO_REPLACE	do replace audio signal
AGS_SFZ_LOADER_HAS_COMPLETED	has completed
AGS_SFZ_LOADER_RUN_APPLY_SYNTH	run apply synth

AGS_TYPE_SFZ_LOADER

```
#define AGS_TYPE_SFZ_LOADER                 (ags_sfz_loader_get_type())
```

struct AgsSFZLoader

```
struct AgsSFZLoader;
```

struct AgsSFZLoaderClass

```
struct AgsSFZLoaderClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned audio.

Owner: AgsSFZLoader

Flags: Read / Write

Since: 3.0.0

The "audio-container" property

```
"audio-container"          AgsAudioContainer~*
```

The audio container opened.

Owner: AgsSFZLoader

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
"filename"                 char~*
```

The filename to open.

Owner: AgsSFZLoader

Flags: Read / Write

Default value: NULL

Since: 3.0.0

12.10 AgsSFZInstrumentLoader

AgsSFZInstrumentLoader — load Soundfont2 MIDI locale asynchronously

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_INSTRUMENT_LOADER_GET_OBJ_MUTEX()
gboolean	ags_sfz_instrument_loader_test_flags ()
void	ags_sfz_instrument_loader_set_flags ()
void	ags_sfz_instrument_loader_unset_flags ()
AgsAudio *	ags_sfz_instrument_loader_get_audio ()
void	ags_sfz_instrument_loader_set_audio ()
gchar *	ags_sfz_instrument_loader_get_filename ()
void	ags_sfz_instrument_loader_set_filename ()
AgsAudioContainer *	ags_sfz_instrument_loader_get_audio_container ()
void	ags_sfz_instrument_loader_set_audio_container ()
AgsSFZSynthUtil *	ags_sfz_instrument_loader_get_template ()
void	ags_sfz_instrument_loader_set_template ()

AgsSFZSynthUtil *	ags_sfz_instrument_loader_get_synth ()
void	ags_sfz_instrument_loader_set_synth ()
void	ags_sfz_instrument_loader_start ()
AgsSFZInstrumentLoader *	ags_sfz_instrument_loader_new ()
#define	AGS_IS_SFZ_INSTRUMENT_LOADER()
#define	AGS_IS_SFZ_INSTRUMENT_LOADER_CLASS()
#define	AGS_SFZ_INSTRUMENT_LOADER()
#define	AGS_SFZ_INSTRUMENT_LOADER_CLASS()
#define	AGS_SFZ_INSTRUMENT_LOADER_GET_CLASS()
GType	ags_sfz_instrument_loader_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsAudioContainer *	audio-container	Read / Write
char *	filename	Read / Write
gpointer	synth	Read / Write
gpointer	template	Read / Write

Types and Values

enum	AgsSFZInstrumentLoaderFlags
#define	AGS_TYPE_SFZ_INSTRUMENT_LOADER
struct	AgsSFZInstrumentLoader
struct	AgsSFZInstrumentLoaderClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsSFZInstrumentLoader
```

Includes

```
#include <ags/plugin/ags_sfz_instrument_loader.h>
```

Description

The AgsSFZInstrumentLoader loads Soundfont2 audio data of MIDI locale asynchronously.

Functions

AGS_SFZ_INSTRUMENT_LOADER_GET_OBJ_MUTEX()

```
#define AGS_SFZ_INSTRUMENT_LOADER_GET_OBJ_MUTEX(obj) (&(((AgsSFZInstrumentLoader *) obj)-> ↩
    obj_mutex))
```

ags_sfz_instrument_loader_test_flags ()

```
gboolean
ags_sfz_instrument_loader_test_flags (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                      guint flags);
```

Test *flags* to be set on *recall* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.17.0

ags_sfz_instrument_loader_set_flags ()

```
void
ags_sfz_instrument_loader_set_flags (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                     guint flags);
```

Set flags.

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
flags	the flags	

Since: 3.17.0

ags_sfz_instrument_loader_unset_flags ()

```
void
ags_sfz_instrument_loader_unset_flags (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                       guint flags);
```

Unset flags.

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
flags	the flags	

Since: 3.17.0

ags_sfz_instrument_loader_get_audio ()

```
AgsAudio~*
ags_sfz_instrument_loader_get_audio (AgsSFZInstrumentLoader *sfz_instrument_loader);
```

Get audio of *sfz_instrument_loader*.

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	

Returns

the assigned AgsAudio.

[transfer full]

Since: 3.17.0

ags_sfz_instrument_loader_set_audio ()

```
void
ags_sfz_instrument_loader_set_audio (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                     AgsAudio *audio);
```

Set audio of *sfz_instrument_loader*.

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
audio	the AgsAudio	

Since: 3.17.0

ags_sfz_instrument_loader_get_filename ()

```
gchar~*
ags_sfz_instrument_loader_get_filename
                                (AgsSFZInstrumentLoader *sfz_instrument_loader);
```

Get filename of *sfz_instrument_loader*.

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	

Returns

the assigned filename.

[transfer full]

Since: 3.17.0

ags_sfz_instrument_loader_set_filename ()

```
void
ags_sfz_instrument_loader_set_filename
                                (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                 gchar *filename);
```

Set filename of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
filename	the filename	

Since: 3.17.0

ags_sfz_instrument_loader_get_audio_container ()

```
AgsAudioContainer~*
ags_sfz_instrument_loader_get_audio_container
                                (AgsSFZInstrumentLoader *sfz_instrument_loader);
```

Get audio container of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	

Returns

the assigned AgsAudioContainer.

[transfer full]

Since: 3.17.0

ags_sfz_instrument_loader_set_audio_container ()

```
void
ags_sfz_instrument_loader_set_audio_container
                                (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                 AgsAudioContainer *audio_container);
```

Set audio container of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
audio_container	the AgsAudioContainer	

Since: 3.17.0

ags_sfz_instrument_loader_get_template ()

```
AgsSFZSynthUtil~*
ags_sfz_instrument_loader_get_template
                                (AgsSFZInstrumentLoader *sfz_instrument_loader);
```

Get AgsSFZSynthUtil template of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader

Returns

the assigned AgsSFZSynthUtil

Since: 3.17.0

ags_sfz_instrument_loader_set_template ()

```
void
ags_sfz_instrument_loader_set_template
                                (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                 AgsSFZSynthUtil *template);
```

Set AgsSFZSynthUtil template of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader
template	the AgsSFZSynthUtil

Since: 3.17.0

ags_sfz_instrument_loader_get_synth ()

```
AgsSFZSynthUtil~*
ags_sfz_instrument_loader_get_synth (AgsSFZInstrumentLoader *sfz_instrument_loader);
```

Get AgsSFZSynthUtil synth of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader

Returns

the assigned AgsSFZSynthUtil

Since: 3.17.0

ags_sfz_instrument_loader_set_synth ()

```
void
ags_sfz_instrument_loader_set_synth (AgsSFZInstrumentLoader *sfz_instrument_loader,
                                     AgsSFZSynthUtil *synth);
```

Set AgsSFZSynthUtil synth of *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	
synth	the AgsSFZSynthUtil	

Since: 3.17.0

ags_sfz_instrument_loader_start ()

```
void
ags_sfz_instrument_loader_start (AgsSFZInstrumentLoader *sfz_instrument_loader);
```

Start *sfz_instrument_loader* .

Parameters

sfz_instrument_loader	the AgsSFZInstrumentLoader	

Since: 3.17.0

ags_sfz_instrument_loader_new ()

```
AgsSFZInstrumentLoader~*
ags_sfz_instrument_loader_new (AgsAudio *audio,
                               gchar *filename);
```

Create a new instance of AgsSFZInstrumentLoader.

Parameters

audio	the AgsAudio	
filename	the filename	

Returns

the new AgsSFZInstrumentLoader

Since: 3.17.0

AGS_IS_SFZ_INSTRUMENT_LOADER()

```
#define AGS_IS_SFZ_INSTRUMENT_LOADER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SFZ_INSTRUMENT_LOADER))
```

AGS_IS_SFZ_INSTRUMENT_LOADER_CLASS()

```
#define AGS_IS_SFZ_INSTRUMENT_LOADER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SFZ_INSTRUMENT_LOADER))
```

AGS_SFZ_INSTRUMENT_LOADER()

```
#define AGS_SFZ_INSTRUMENT_LOADER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_INSTRUMENT_LOADER, AgsSFZInstrumentLoader))
```

AGS_SFZ_INSTRUMENT_LOADER_CLASS()

```
#define AGS_SFZ_INSTRUMENT_LOADER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_INSTRUMENT_LOADER, AgsSFZInstrumentLoaderClass))
```

AGS_SFZ_INSTRUMENT_LOADER_GET_CLASS()

```
#define AGS_SFZ_INSTRUMENT_LOADER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SFZ_INSTRUMENT_LOADER, AgsSFZInstrumentLoaderClass))
```

ags_sfz_instrument_loader_get_type ()

```
GType
ags_sfz_instrument_loader_get_type ();
```

Types and Values

enum AgsSFZInstrumentLoaderFlags

Enum values to configure Soundfont2 MIDI locale loader.

Members

AGS_SFZ_INSTRUMENT_LOADER_HAS_COMPLETED	has completed
AGS_SFZ_INSTRUMENT_LOADER_RUN_APPLY_INSTRUMENT	

AGS_TYPE_SFZ_INSTRUMENT_LOADER

```
#define AGS_TYPE_SFZ_INSTRUMENT_LOADER                 (ags_sfz_instrument_loader_get_type() ↩
    )
```

struct AgsSFZInstrumentLoader

```
struct AgsSFZInstrumentLoader;
```

struct AgsSFZInstrumentLoaderClass

```
struct AgsSFZInstrumentLoaderClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned audio.

Owner: AgsSFZInstrumentLoader

Flags: Read / Write

Since: 3.17.0

The "audio-container" property

```
  "audio-container"           AgsAudioContainer~*
```

The audio container opened.

Owner: AgsSFZInstrumentLoader

Flags: Read / Write

Since: 3.17.0

The "filename" property

```
  "filename"                  char~*
```

The filename to open.

Owner: AgsSFZInstrumentLoader

Flags: Read / Write

Default value: NULL

Since: 3.17.0

The "synth" property

```
  "synth"                     gpointer
```

The assigned synth.

Owner: AgsSFZInstrumentLoader

Flags: Read / Write

Since: 3.17.0

The "template" property

```
"template"                  gpointer
```

The assigned template.

Owner: AgsSFZInstrumentLoader

Flags: Read / Write

Since: 3.17.0

12.11 AgsWaveLoader

AgsWaveLoader — load wave asynchronously

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_WAVE_LOADER_GET_OBJ_MUTEX()
gboolean	ags_wave_loader_test_flags ()
void	ags_wave_loader_set_flags ()
void	ags_wave_loader_unset_flags ()
AgsAudio *	ags_wave_loader_get_audio ()
void	ags_wave_loader_set_audio ()
gchar *	ags_wave_loader_get_filename ()
void	ags_wave_loader_set_filename ()
AgsAudioFile *	ags_wave_loader_get_audio_file ()
void	ags_wave_loader_set_audio_file ()
void	ags_wave_loader_start ()
AgsWaveLoader *	ags_wave_loader_new ()
#define	AGS_IS_WAVE_LOADER()
#define	AGS_IS_WAVE_LOADER_CLASS()
#define	AGS_WAVE_LOADER()
#define	AGS_WAVE_LOADER_CLASS()
#define	AGS_WAVE_LOADER_GET_CLASS()
GType	ags_wave_loader_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsAudioFile *	audio-file	Read / Write
char *	filename	Read / Write

Types and Values

enum	AgsWaveLoaderFlags
#define	AGS_TYPE_WAVE_LOADER

struct	AgsWaveLoader
struct	AgsWaveLoaderClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsWaveLoader
```

Includes

```
#include <ags/plugin/ags_wave_loader.h>
```

Description

The AgsWaveLoader loads wave audio data asynchronously.

Functions

AGS_WAVE_LOADER_GET_OBJ_MUTEX()

```
#define AGS_WAVE_LOADER_GET_OBJ_MUTEX(obj) (&(((AgsWaveLoader *) obj)->obj_mutex))
```

ags_wave_loader_test_flags ()

```
gboolean
ags_wave_loader_test_flags (AgsWaveLoader *wave_loader,
                            guint flags);
```

Test *flags* to be set on *recall* .

Parameters

wave_loader	the AgsWaveLoader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_wave_loader_set_flags ()

```
void
ags_wave_loader_set_flags (AgsWaveLoader *wave_loader,
                           guint flags);
```

Set flags.

Parameters

wave_loader	the AgsWaveLoader	
flags	the flags	

Since: 3.0.0

ags_wave_loader_unset_flags ()

```
void
ags_wave_loader_unset_flags (AgsWaveLoader *wave_loader,
                             guint flags);
```

Unset flags.

Parameters

wave_loader	the AgsWaveLoader	
flags	the flags	

Since: 3.0.0

ags_wave_loader_get_audio ()

```
AgsAudio~*
ags_wave_loader_get_audio (AgsWaveLoader *wave_loader);
```

Get audio of *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	

Returns

the assigned AgsAudio.

[transfer full]

Since: 3.2.0

ags_wave_loader_set_audio ()

```
void
ags_wave_loader_set_audio (AgsWaveLoader *wave_loader,
                           AgsAudio *audio);
```

Set audio of *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	
audio	the AgsAudio	

Since: 3.2.0

ags_wave_loader_get_filename ()

```
gchar~*
ags_wave_loader_get_filename (AgsWaveLoader *wave_loader);
```

Get filename of *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	

Returns

the assigned filename

Since: 3.2.0

ags_wave_loader_set_filename ()

```
void
ags_wave_loader_set_filename (AgsWaveLoader *wave_loader,
                              gchar *filename);
```

Set filename of *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	
filename	the filename	

Since: 3.2.0

ags_wave_loader_get_audio_file ()

```
AgsAudioFile~*
ags_wave_loader_get_audio_file (AgsWaveLoader *wave_loader);
```

Get audio file of *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	

Returns

the assigned AgsAudioFile.

[transfer full]

Since: 3.2.0

ags_wave_loader_set_audio_file ()

```
void
ags_wave_loader_set_audio_file (AgsWaveLoader *wave_loader,
                                AgsAudioFile *audio_file);
```

Set audio file of *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	
audio_file	the AgsAudioFile	

Since: 3.2.0

ags_wave_loader_start ()

```
void
ags_wave_loader_start (AgsWaveLoader *wave_loader);
```

Start *wave_loader* .

Parameters

wave_loader	the AgsWaveLoader	

Since: 3.0.0

ags_wave_loader_new ()

```
AgsWaveLoader~*
ags_wave_loader_new (AgsAudio *audio,
                     gchar *filename,
                     gboolean do_replace);
```

Create a new instance of AgsWaveLoader.

Parameters

audio	the AgsAudio	
filename	the filename	
do_replace	if TRUE replace *audio* 's wave	

Returns

the new AgsWaveLoader

Since: 3.0.0

AGS_IS_WAVE_LOADER()

```
#define AGS_IS_WAVE_LOADER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_WAVE_LOADER))
```

AGS_IS_WAVE_LOADER_CLASS()

```
#define AGS_IS_WAVE_LOADER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_WAVE_LOADER))
```

AGS_WAVE_LOADER()

```
#define AGS_WAVE_LOADER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_WAVE_LOADER, AgsWaveLoader))
```

AGS_WAVE_LOADER_CLASS()

```
#define AGS_WAVE_LOADER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_WAVE_LOADER, AgsWaveLoaderClass))
```

AGS_WAVE_LOADER_GET_CLASS()

```
#define AGS_WAVE_LOADER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_WAVE_LOADER, AgsWaveLoaderClass))
```

ags_wave_loader_get_type ()

```
GType
ags_wave_loader_get_type ();
```

Types and Values

enum AgsWaveLoaderFlags

Members

AGS_WAVE_LOADER_DO_REPLACE	
AGS_WAVE_LOADER_HAS_COMPLETED	

AGS_TYPE_WAVE_LOADER

```
#define AGS_TYPE_WAVE_LOADER                (ags_wave_loader_get_type())
```

struct AgsWaveLoader

```
struct AgsWaveLoader;
```

struct AgsWaveLoaderClass

```
struct AgsWaveLoaderClass {
  GObjectClass gobject;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned audio.

Owner: AgsWaveLoader

Flags: Read / Write

Since: 3.0.0

The "audio-file" property

```
  "audio-file"                AgsAudioFile~*
```

The audio file opened.

Owner: AgsWaveLoader

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
  "filename"                  char~*
```

The filename to open.

Owner: AgsWaveLoader

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Chapter 13

Audio MIDI - MIDI support

Objects related to MIDI disk IO and parsing.

13.1 AgsMidiBufferUtil

AgsMidiBufferUtil — MIDI buffer util

Stability Level

Stable, unless otherwise indicated

Functions

guint	ags_midi_buffer_util_get_varlength_size ()
void	ags_midi_buffer_util_put_varlength ()
guint	ags_midi_buffer_util_get_varlength ()
void	ags_midi_buffer_util_put_int16 ()
void	ags_midi_buffer_util_get_int16 ()
void	ags_midi_buffer_util_put_int24 ()
void	ags_midi_buffer_util_get_int24 ()
void	ags_midi_buffer_util_put_int32 ()
void	ags_midi_buffer_util_get_int32 ()
void	ags_midi_buffer_util_put_header ()
guint	ags_midi_buffer_util_get_header ()
void	ags_midi_buffer_util_put_track ()
guint	ags_midi_buffer_util_get_track ()
void	ags_midi_buffer_util_put_key_on ()
guint	ags_midi_buffer_util_get_key_on ()
void	ags_midi_buffer_util_put_key_off ()
guint	ags_midi_buffer_util_get_key_off ()
void	ags_midi_buffer_util_put_key_pressure ()
guint	ags_midi_buffer_util_get_key_pressure ()
void	ags_midi_buffer_util_put_change_parameter ()
guint	ags_midi_buffer_util_get_change_parameter ()
void	ags_midi_buffer_util_put_pitch_bend ()
guint	ags_midi_buffer_util_get_pitch_bend ()
void	ags_midi_buffer_util_put_change_program ()

guint	ags_midi_buffer_util_get_change_program ()
void	ags_midi_buffer_util_put_change_pressure ()
guint	ags_midi_buffer_util_get_change_pressure ()
void	ags_midi_buffer_util_put_sysex ()
guint	ags_midi_buffer_util_get_sysex ()
void	ags_midi_buffer_util_put_quarter_frame ()
guint	ags_midi_buffer_util_get_quarter_frame ()
void	ags_midi_buffer_util_put_song_position ()
guint	ags_midi_buffer_util_get_song_position ()
void	ags_midi_buffer_util_put_song_select ()
guint	ags_midi_buffer_util_get_song_select ()
void	ags_midi_buffer_util_put_tune_request ()
guint	ags_midi_buffer_util_get_tune_request ()
void	ags_midi_buffer_util_put_sequence_number ()
guint	ags_midi_buffer_util_get_sequence_number ()
void	ags_midi_buffer_util_put_smtpe ()
guint	ags_midi_buffer_util_get_smtpe ()
void	ags_midi_buffer_util_put_tempo ()
guint	ags_midi_buffer_util_get_tempo ()
void	ags_midi_buffer_util_put_time_signature ()
guint	ags_midi_buffer_util_get_time_signature ()
void	ags_midi_buffer_util_put_key_signature ()
guint	ags_midi_buffer_util_get_key_signature ()
void	ags_midi_buffer_util_put_sequencer_meta_event ()
guint	ags_midi_buffer_util_get_sequencer_meta_event ()
void	ags_midi_buffer_util_put_text_event ()
guint	ags_midi_buffer_util_get_text_event ()
void	ags_midi_buffer_util_put_end_of_track ()
guint	ags_midi_buffer_util_get_end_of_track ()
guchar *	ags_midi_buffer_util_seek_message ()
guint	ags_midi_buffer_util_decode ()
GType	ags_midi_buffer_util_get_type ()

Types and Values

enum	AgsMidiBufferUtilMtcQuarterFrameMessageType
enum	AgsMidiBufferUtilSmtpeFrameRate
#define	AGS_TYPE_MIDI_BUFFER_UTIL
struct	AgsMidiBufferUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsMidiBufferUtil
```

Includes

```
#include <ags/audio/midi/ags_midi_buffer_util.h>
```

Description

Utility functions for MIDI buffers.

Functions

ags_midi_buffer_util_get_varlength_size ()

```
guint
ags_midi_buffer_util_get_varlength_size
                                (glong varlength);
```

Retrieve the size needed to store the variable length.

Parameters

varlength	the variable length	

Returns

the size needed to be allocated

Since: 3.0.0

ags_midi_buffer_util_put_varlength ()

```
void
ags_midi_buffer_util_put_varlength (guchar *buffer,
                                    glong varlength);
```

Put the variable lenght value to *buffer* .

Parameters

buffer	the character buffer	
varlength	the value to put	

Since: 3.0.0

ags_midi_buffer_util_get_varlength ()

```
guint
ags_midi_buffer_util_get_varlength (guchar *buffer,
                                    glong *varlength);
```

Get the variable lenght value from *buffer* .

Parameters

buffer	the character buffer	
varlength	the return location	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_int16 ()

```
void
ags_midi_buffer_util_put_int16 (guchar *buffer,
                                glong val);
```

Put signed 16 bit integer.

Parameters

buffer	the character buffer	
val	the integer	

Since: 3.0.0

ags_midi_buffer_util_get_int16 ()

```
void
ags_midi_buffer_util_get_int16 (guchar *buffer,
                                glong *val);
```

Get signed 32 bit integer.

Parameters

buffer	the character buffer	
val	return location of the integer	

Since: 3.0.0

ags_midi_buffer_util_put_int24 ()

```
void
ags_midi_buffer_util_put_int24 (guchar *buffer,
                                glong val);
```

Put signed 24 bit integer.

Parameters

buffer	the character buffer	
val	the integer	

Since: 3.0.0

ags_midi_buffer_util_get_int24 ()

```
void
ags_midi_buffer_util_get_int24 (guchar *buffer,
                                glong *val);
```

Get signed 24 bit integer.

Parameters

buffer	the character buffer	
val	return location of the integer	

Since: 3.0.0

ags_midi_buffer_util_put_int32 ()

```
void
ags_midi_buffer_util_put_int32 (guchar *buffer,
                                glong val);
```

Put signed 32 bit integer.

Parameters

buffer	the character buffer	
val	the integer	

Since: 3.0.0

ags_midi_buffer_util_get_int32 ()

```
void
ags_midi_buffer_util_get_int32 (guchar *buffer,
                                glong *val);
```

Get signed 32 bit integer.

Parameters

buffer	the character buffer	
val	return location of the integer	

Since: 3.0.0

ags_midi_buffer_util_put_header ()

```
void
ags_midi_buffer_util_put_header (guchar *buffer,
                                 glong offset,
                                 glong format,
                                 glong track_count,
                                 glong division);
```

Puts the midi header.

Parameters

buffer	the character buffer	
offset	start delta-time	
format	either 0, 1 or 2.	
track_count	the number of tracks	
division	timing division	

Since: 3.0.0

ags_midi_buffer_util_get_header ()

```
guint
ags_midi_buffer_util_get_header (guchar *buffer,
                                 glong *offset,
                                 glong *format,
                                 glong *track_count,
                                 glong *division);
```

Gets the midi header

Parameters

buffer	the character buffer	
offset	start delta-time	
format	either 0, 1 or 2.	
track_count	the number of tracks	
division	timing division	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_track ()

```
void
ags_midi_buffer_util_put_track (guchar *buffer,
                                glong offset);
```

Put track.

Parameters

buffer	the midi buffer	
offset	start delta-time	

Since: 3.0.0

ags_midi_buffer_util_get_track ()

```
guint
ags_midi_buffer_util_get_track (guchar *buffer,
                                glong *offset);
```

Get track

Parameters

buffer	the midi buffer	
offset	start delta-time	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_key_on ()

```
void
ags_midi_buffer_util_put_key_on (guchar *buffer,
                                 glong delta_time,
                                 glong channel,
                                 glong key,
                                 glong velocity);
```

Puts the given values to *buffer* with appropriate channel message.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	valid channels from 0-15	
key	valid keys to play 0-128	
velocity	the key dynamics	

Since: 3.0.0

ags_midi_buffer_util_get_key_on ()

```
guint
ags_midi_buffer_util_get_key_on (guchar *buffer,
                                 glong *delta_time,
                                 glong *channel,
                                 glong *key,
                                 glong *velocity);
```

Get the key-on message from *buffer* .

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
channel	the return location of channel	
key	the return location of key	
velocity	the return location of velocity	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_key_off ()

```
void
ags_midi_buffer_util_put_key_off (guchar *buffer,
                                  glong delta_time,
                                  glong channel,
                                  glong key,
                                  glong velocity);
```

Puts the given values to *buffer* with appropriate channel message.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	valid channels from 0-15	
key	valid keys to play 0-128	
velocity	the key dynamics	

Since: 3.0.0

ags_midi_buffer_util_get_key_off ()

```
guint
ags_midi_buffer_util_get_key_off (guchar *buffer,
                                  glong *delta_time,
                                  glong *channel,
                                  glong *key,
                                  glong *velocity);
```

Get the key-off message from *buffer* .

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
channel	the return location of channel	
key	the return location of key	
velocity	the return location of velocity	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_key_pressure ()

```
void
ags_midi_buffer_util_put_key_pressure (guchar *buffer,
                                       glong delta_time,
                                       glong channel,
                                       glong key,
                                       glong pressure);
```

Puts the given values to *buffer* with appropriate channel message.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	valid channels from 0-15	
key	valid keys to play 0-128	
pressure	the key dynamics	

Since: 3.0.0

ags_midi_buffer_util_get_key_pressure ()

```
guint
ags_midi_buffer_util_get_key_pressure (guchar *buffer,
                                       glong *delta_time,
                                       glong *channel,
                                       glong *key,
                                       glong *pressure);
```

Get the key-pressure message from *buffer* .

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
channel	the return location of channel	
key	the return location of key	
pressure	the return location of pressure	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_change_parameter ()

```
void
ags_midi_buffer_util_put_change_parameter
                                (guchar *buffer,
                                 glong delta_time,
                                 glong channel,
                                 glong control,
                                 glong value);
```

Put change parameter.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	channel	
control	the control	
value	the value	

Since: 3.0.0

ags_midi_buffer_util_get_change_parameter ()

```
guint
ags_midi_buffer_util_get_change_parameter
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *channel,
                                 glong *control,
                                 glong *value);
```

Get change parameter.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
channel	the return location of channel	
control	the return location of the control	
value	the return location the value	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_pitch_bend ()

```
void
ags_midi_buffer_util_put_pitch_bend (guchar *buffer,
                                     glong delta_time,
                                     glong channel,
                                     glong pitch,
                                     glong transmitter);
```

Put pitch bend.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	channel	
pitch	the pitch	
transmitter	the transmitter	

Since: 3.0.0

ags_midi_buffer_util_get_pitch_bend ()

```
guint
ags_midi_buffer_util_get_pitch_bend (guchar *buffer,
                                     glong *delta_time,
                                     glong *channel,
                                     glong *pitch,
                                     glong *transmitter);
```

Get pitch bend.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
channel	the return location of channel	
pitch	the return location of the pitch	
transmitter	the return location the transmitter	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_change_program ()

```
void
ags_midi_buffer_util_put_change_program
                                (guchar *buffer,
                                 glong delta_time,
                                 glong channel,
                                 glong program);
```

Put change program.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	channel	
program	the program	

Since: 3.0.0

ags_midi_buffer_util_get_change_program ()

```
guint
ags_midi_buffer_util_get_change_program
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *channel,
                                 glong *program);
```

Get change program.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
channel	the return location of channel	
program	the return location of the program	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_change_pressure ()

```
void
ags_midi_buffer_util_put_change_pressure
                                (guchar *buffer,
                                 glong delta_time,
                                 glong channel,
                                 glong pressure);
```

Put change pressure.

Parameters

buffer	the character buffer	
delta_time	timing information	
channel	channel	
pressure	the pressure	

Since: 3.0.0

ags_midi_buffer_util_get_change_pressure ()

```
guint
ags_midi_buffer_util_get_change_pressure
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *channel,
                                 glong *pressure);
```

Get change pressure.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	

channel	the return location of channel	
pressure	the return location of the pressure	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_sysex ()

```
void
ags_midi_buffer_util_put_sysex (guchar *buffer,
                                glong delta_time,
                                guchar *data,
                                glong length);
```

Put sysex.

Parameters

buffer	the character buffer	
delta_time	timing information	
data	the data	
length	the data's length	

Since: 3.0.0

ags_midi_buffer_util_get_sysex ()

```
guint
ags_midi_buffer_util_get_sysex (guchar *buffer,
                                glong *delta_time,
                                guchar **data,
                                glong *length);
```

Get sysex.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
data	the return location of data	
length	the return location of length	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_quarter_frame ()

```
void
ags_midi_buffer_util_put_quarter_frame
                                (guchar *buffer,
                                 glong delta_time,
                                 glong message_type,
                                 glong values);
```

Put quarter frame.

Parameters

buffer	the character buffer	
delta_time	timing information	
message_type	the message type	
values	the values	

Since: 3.0.0

ags_midi_buffer_util_get_quarter_frame ()

```
guint
ags_midi_buffer_util_get_quarter_frame
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *message_type,
                                 glong *values);
```

Get quarter frame.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
message_type	the return location of the message type	
values	the return location of the values	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_song_position ()

```
void
ags_midi_buffer_util_put_song_position
                                (guchar *buffer,
                                 glong delta_time,
                                 glong song_position);
```

Put song position.

Parameters

buffer	the character buffer	
delta_time	timing information	
song_position	the song position	

Since: 3.0.0

ags_midi_buffer_util_get_song_position ()

```
guint
ags_midi_buffer_util_get_song_position
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *song_position);
```

Get song position.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
song_position	the return location of the song position	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_song_select ()

```
void
ags_midi_buffer_util_put_song_select (guchar *buffer,
                                      glong delta_time,
                                      glong song_select);
```

Put song select.

Parameters

buffer	the character buffer	
delta_time	timing information	
song_select	the song select	

Since: 3.0.0

ags_midi_buffer_util_get_song_select ()

```
guint
ags_midi_buffer_util_get_song_select (guchar *buffer,
                                      glong *delta_time,
                                      glong *song_select);
```

Get song select.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
song_select	the return location of the song select	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_tune_request ()

```
void
ags_midi_buffer_util_put_tune_request (guchar *buffer,
                                       glong delta_time);
```

Put tune request

Parameters

buffer	the character buffer	
delta_time	timing information	

Since: 3.0.0

ags_midi_buffer_util_get_tune_request ()

```
guint
ags_midi_buffer_util_get_tune_request (guchar *buffer,
                                       glong *delta_time);
```

Get tune request.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_sequence_number ()

```
void
ags_midi_buffer_util_put_sequence_number
                                (guchar *buffer,
                                 glong delta_time,
                                 glong sequence);
```

Put sequence number.

Parameters

buffer	the character buffer	
delta_time	timing information	
sequence	the sequence	

Since: 3.0.0

ags_midi_buffer_util_get_sequence_number ()

```
guint
ags_midi_buffer_util_get_sequence_number
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *sequence);
```

Get sequence number.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
sequence	the return location of the sequence	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_smtpe ()

```
void
ags_midi_buffer_util_put_smtpe (guchar *buffer,
                                glong delta_time,
                                glong rr,
                                glong hr,
                                glong mn,
                                glong se,
                                glong fr);
```

Put smtpe timestamp.

Parameters

buffer	the character buffer	
delta_time	timing information	
rr	frame rate	
hr	hour	
mn	minute	
se	second	
fr	frame number	

Since: 3.0.0

ags_midi_buffer_util_get_smtpe ()

```
guint
ags_midi_buffer_util_get_smtpe (guchar *buffer,
                                glong *delta_time,
                                glong *rr,
                                glong *hr,
                                glong *mn,
                                glong *se,
                                glong *fr);
```

Get smtpe timestamp.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
rr	the return location of frame rate	
hr	the return location of hour	
mn	the return location of minute	
se	the return location of second	
fr	the return location of frame number	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_tempo ()

```
void
ags_midi_buffer_util_put_tempo (guchar *buffer,
                                glong delta_time,
                                glong tempo);
```

Put tempo.

Parameters

buffer	the character buffer	
delta_time	timing information	
tempo	the tempo	

Since: 3.0.0

ags_midi_buffer_util_get_tempo ()

```
guint
ags_midi_buffer_util_get_tempo (guchar *buffer,
                                glong *delta_time,
                                glong *tempo);
```

Get tempo.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
tempo	the tempo	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_time_signature ()

```
void
ags_midi_buffer_util_put_time_signature
                               (guchar *buffer,
                                glong delta_time,
                                glong nn,
                                glong dd,
                                glong cc,
                                glong bb);
```

Put time signature

Parameters

buffer	the character buffer	
delta_time	timing information	
nn	numerator	
dd	denominator	
cc	clocks	
bb	beats	

Since: 3.0.0

ags_midi_buffer_util_get_time_signature ()

```
guint
ags_midi_buffer_util_get_time_signature
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *nn,
                                 glong *dd,
                                 glong *cc,
                                 glong *bb);
```

Get time signature.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
nn	the return location of numerator	
dd	the return location of denominator	
cc	the return location of clocks	
bb	the return location of beats	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_key_signature ()

```
void
ags_midi_buffer_util_put_key_signature
                                (guchar *buffer,
                                 glong delta_time,
                                 glong sf,
                                 glong mi);
```

Put key signature.

Parameters

buffer	the character buffer	
delta_time	timing information	
sf	flats or sharps	
mi	1 equals minor or 0 means major	

Since: 3.0.0

ags_midi_buffer_util_get_key_signature ()

```
guint
ags_midi_buffer_util_get_key_signature
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *sf,
                                 glong *mi);
```

Get key signature.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
sf	the return location of flats or sharps	
mi	the return location of minor or major	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_sequencer_meta_event ()

```
void
ags_midi_buffer_util_put_sequencer_meta_event
                                (guchar *buffer,
                                 glong delta_time,
                                 glong len,
                                 glong id,
                                 glong data);
```

Put sequencer meta event.

Parameters

buffer	the character buffer	
delta_time	timing information	
len	the length of data	
id	the manufacturer id	
data	the data	

Since: 3.0.0

ags_midi_buffer_util_get_sequencer_meta_event ()

```
guint
ags_midi_buffer_util_get_sequencer_meta_event
                                (guchar *buffer,
                                 glong *delta_time,
                                 glong *len,
                                 glong *id,
                                 glong *data);
```

Get sequencer meta event.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
len	the return location of the length of data	
id	the return location of the manufacturer id	
data	the return location of the data	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_text_event ()

```
void
ags_midi_buffer_util_put_text_event (guchar *buffer,
                                     glong delta_time,
                                     gchar *text,
                                     glong length);
```

Put text event.

Parameters

buffer	the character buffer	
delta_time	timing information	
text	the text	
length	the length	

Since: 3.0.0

ags_midi_buffer_util_get_text_event ()

```
guint
ags_midi_buffer_util_get_text_event (guchar *buffer,
                                     glong *delta_time,
                                     gchar **text,
                                     glong *length);
```

Get text event.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	
text	the return location of the text	
length	the return location of the length	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_put_end_of_track ()

```
void
ags_midi_buffer_util_put_end_of_track (guchar *buffer,
                                       glong delta_time);
```

Put end of track.

Parameters

buffer	the character buffer	
delta_time	timing information	

Since: 3.0.0

ags_midi_buffer_util_get_end_of_track ()

```
guint
ags_midi_buffer_util_get_end_of_track (guchar *buffer,
                                       glong *delta_time);
```

Get end of track.

Parameters

buffer	the character buffer	
delta_time	the return location of timing information	

Returns

the number of bytes read.

Since: 3.0.0

ags_midi_buffer_util_seek_message ()

```
guchar~*
ags_midi_buffer_util_seek_message (guchar *buffer,
                                   guint message_count,
                                   glong *delta_time);
```

Seek MIDI messages from *buffer*

Parameters

buffer	the buffer to seek	
message_count	seek count messages	
delta_time	the return location of current delta time	

Returns

the buffer at offset at *message_count* ahead

Since: 3.0.0

ags_midi_buffer_util_decode ()

```
guint
ags_midi_buffer_util_decode (guchar *buffer,
                             snd_seq_event_t *event);
```

Decode *event* to *buffer*

Parameters

buffer	the midi buffer	
event	the ALSA sequencer event	

Returns

the bytes written

Since: 3.0.0

ags_midi_buffer_util_get_type ()

```
GType
ags_midi_buffer_util_get_type (void);
```

Types and Values

enum AgsMidiBufferUtilMtcQuarterFrameMessageType

Members

AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_FRAME_NUMBER_LSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_FRAME_NUMBER_MSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_SECONDS_LSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_SECONDS_MSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_MINUTES_LSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_MINUTES_MSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_HOURS_LSB	
AGS_MIDI_BUFFER_UTIL_MTC_QUARTER_FRAME_HOURS_MSB	

enum AgsMidiBufferUtilSmtpeFrameRate

Members

AGS_MIDI_BUFFER_UTIL_SMTPE_FRAME_RATE_24_FPS	
AGS_MIDI_BUFFER_UTIL_SMTPE_FRAME_RATE_25_FPS	
AGS_MIDI_BUFFER_UTIL_SMTPE_FRAME_RATE_30_FPS	
AGS_MIDI_BUFFER_UTIL_SMTPE_FRAME_RATE_30_FPS_DROP_FRAME	

AGS_TYPE_MIDI_BUFFER_UTIL

```
#define AGS_TYPE_MIDI_BUFFER_UTIL          (ags_midi_buffer_util_get_type())
```

struct AgsMidiBufferUtil

```
struct AgsMidiBufferUtil {
  //empty
};
```

13.2 AgsMidiFile

AgsMidiFile — the MIDI file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MIDI_FILE_GET_OBJ_MUTEX()
#define	AGS_MIDI_FILE_TRACK()
GQuark	ags_midi_file_error_quark ()
gboolean	ags_midi_file_open ()
gboolean	ags_midi_file_open_from_data ()
gboolean	ags_midi_file_rw_open ()
void	ags_midi_file_close ()
unsigned char *	ags_midi_file_read ()
void	ags_midi_file_write ()
void	ags_midi_file_seek ()
void	ags_midi_file_flush ()
unsigned char	ags_midi_file_read_byte ()
gint16	ags_midi_file_read_gint16 ()
gint32	ags_midi_file_read_gint24 ()
gint32	ags_midi_file_read_gint32 ()
long	ags_midi_file_read_varlength ()
unsigned char *	ags_midi_file_read_text ()
void	ags_midi_file_write_byte ()
void	ags_midi_file_write_gint16 ()
void	ags_midi_file_write_gint24 ()
void	ags_midi_file_write_gint32 ()
void	ags_midi_file_write_varlength ()
void	ags_midi_file_write_text ()
unsigned char *	ags_midi_file_read_header ()
void	ags_midi_file_write_header ()
unsigned char *	ags_midi_file_read_track_data ()
void	ags_midi_file_write_track_data ()
void	ags_midi_file_read_notation ()
void	ags_mid_file_read_midi ()
AgsMidiFile *	ags_midi_file_new ()
#define	AGS_IS_MIDI_FILE()
#define	AGS_IS_MIDI_FILE_CLASS()
#define	AGS_MIDI_FILE()
#define	AGS_MIDI_FILE_CLASS()
#define	AGS_MIDI_FILE_GET_CLASS()
GType	ags_midi_file_get_type ()

Properties

char *	filename	Read / Write

Types and Values

#define	AGS_MIDI_FILE_MAX_TEXT_LENGTH
#define	AGS_MIDI_FILE_MTHD
#define	AGS_MIDI_FILE_MTRK
#define	AGS_MIDI_FILE_DEFAULT_OFFSET
#define	AGS_MIDI_FILE_DEFAULT_FORMAT
#define	AGS_MIDI_FILE_DEFAULT_BEATS
#define	AGS_MIDI_FILE_DEFAULT_FPS
#define	AGS_MIDI_FILE_DEFAULT_TICKS

enum	AgsMidiFileStatus
enum	AgsMidiFileFlags
#define	AGS_MIDI_FILE_ERROR
enum	AgsMidiFileError
struct	AgsMidiFileTrack
#define	AGS_TYPE_MIDI_FILE
struct	AgsMidiFile
struct	AgsMidiFileClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsMidiFile
```

Includes

```
#include <ags/audio/midi/ags_midi_file.h>
```

Description

AgsMidiFile reads or writes your midi files.

Functions

AGS_MIDI_FILE_GET_OBJ_MUTEX()

```
#define AGS_MIDI_FILE_GET_OBJ_MUTEX(obj) (&(((AgsMidiFile *) obj)->obj_mutex))
```

AGS_MIDI_FILE_TRACK()

```
#define AGS_MIDI_FILE_TRACK(ptr) ((AgsMidiFileTrack *)(ptr))
```

ags_midi_file_error_quark ()

```
GQuark
ags_midi_file_error_quark ();
```

ags_midi_file_open ()

```
gboolean
ags_midi_file_open (AgsMidiFile *midi_file,
                    gchar *filename);
```

Opens a MIDI file read-only.

Parameters

midi_file	the AgsMidiFile	
filename	the filename	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_file_open_from_data ()

```
gboolean
ags_midi_file_open_from_data (AgsMidiFile *midi_file,
                              unsigned char *data,
                              guint buffer_length);
```

Opens a virtual MIDI file residing in `data` 's array.

Parameters

midi_file	the AgsMidiFile	
data	the buffer to set	
buffer_length	the length of the buffer	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_file_rw_open ()

```
gboolean
ags_midi_file_rw_open (AgsMidiFile *midi_file,
                       gchar *filename,
                       gboolean create);
```

Opens a MIDI file with read-write permission.

Parameters

midi_file	the AgsMidiFile	
filename	the filename	
create	TRUE create file if not exists, else if FALSE return	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_file_close ()

```
void
ags_midi_file_close (AgsMidiFile *midi_file);
```

Closes the file stream.

Parameters

midi_file	the AgsMidiFile

Since: 3.0.0

ags_midi_file_read ()

```
unsigned char~*
ags_midi_file_read (AgsMidiFile *midi_file);
```

Reads all bytes of the file's stream and stores them in the internal buffer.

Parameters

midi_file	the AgsMidiFile

Returns

the data array just read

Since: 3.0.0

ags_midi_file_write ()

```
void
ags_midi_file_write (AgsMidiFile *midi_file,
                     unsigned char *data,
                     guint buffer_length);
```

Writes *data* to the file stream and to internal buffer, reallocates it if necessary.

Parameters

midi_file	the AgsMidiFile	
data	the data array to write	
buffer_length	n-bytes to write	

Since: 3.0.0

ags_midi_file_seek ()

```
void
ags_midi_file_seek (AgsMidiFile *midi_file,
                    guint position,
                    gint whence);
```

Seeks the file stream's offset.

Parameters

midi_file	the AgsMidiFile	
position	the offset	
whence	SEEK_SET, SEEK_END, or SEEK_CUR	

Since: 3.0.0

ags_midi_file_flush ()

```
void
ags_midi_file_flush (AgsMidiFile *midi_file);
```

Flushes file stream's data buffer to disc.

Parameters

midi_file	the AgsMidiFile	

Since: 3.0.0

ags_midi_file_read_byte ()

```
unsigned char
ags_midi_file_read_byte (AgsMidiFile *midi_file,
                         GError **error);
```

Reads a unsigned char quantity.

Parameters

midi_file	the AgsMidiFile	
error	the GError pointer return location	

Returns

the current value at file's iteration pointer

Since: 3.0.0

ags_midi_file_read_gint16 ()

```
gint16
ags_midi_file_read_gint16 (AgsMidiFile *midi_file,
                           GError **error);
```

Reads a gint16 quantity.

Parameters

midi_file	the AgsMidiFile	
error	the GError pointer return location	

Returns

the current value at file's iteration pointer

Since: 3.0.0

ags_midi_file_read_gint24 ()

```
gint32
ags_midi_file_read_gint24 (AgsMidiFile *midi_file,
                           GError **error);
```

Reads a 24-bit quantity.

Parameters

midi_file	the AgsMidiFile	
error	the GError pointer return location	

Returns

the current value at file's iteration pointer

Since: 3.0.0

ags_midi_file_read_gint32 ()

```
gint32
ags_midi_file_read_gint32 (AgsMidiFile *midi_file,
                           GError **error);
```

Reads a gint32 quantity.

Parameters

midi_file	the AgsMidiFile	
error	the GError pointer return location	

Returns

the current value at file's iteration pointer

Since: 3.0.0

ags_midi_file_read_varlength ()

```
long
ags_midi_file_read_varlength (AgsMidiFile *midi_file,
                              GError **error);
```

Reads a variable length quantity.

Parameters

midi_file	the AgsMidiFile	
error	the GError pointer return location	

Returns

the current value at file's iteration pointer

Since: 3.0.0

ags_midi_file_read_text ()

```
unsigned char~*
ags_midi_file_read_text (AgsMidiFile *midi_file,
                         gint length,
                         GError **error);
```

Reads a string.

Parameters

midi_file	the AgsMidiFile	
length	the number of bytes to be read, or as long valid string for -1	
error	the GError pointer return location	

Returns

the string at file's iteration pointer

Since: 3.0.0

ags_midi_file_write_byte ()

```
void
ags_midi_file_write_byte (AgsMidiFile *midi_file,
                          unsigned char val);
```

Writes a unsigned char quantity to internal buffer.

Parameters

midi_file	the AgsMidiFile	
val	the value to write	

Since: 3.0.0

ags_midi_file_write_gint16 ()

```
void
ags_midi_file_write_gint16 (AgsMidiFile *midi_file,
                            gint16 val);
```

Writes a gint16 quantity to internal buffer.

Parameters

midi_file	the AgsMidiFile	
val	the value to write	

Since: 3.0.0

ags_midi_file_write_gint24 ()

```
void
ags_midi_file_write_gint24 (AgsMidiFile *midi_file,
                            gint32 val);
```

Writes a 24-bit quantity to internal buffer.

Parameters

midi_file	the AgsMidiFile	
val	the value to write	

Since: 3.0.0

ags_midi_file_write_gint32 ()

```
void
ags_midi_file_write_gint32 (AgsMidiFile *midi_file,
                            gint32 val);
```

Writes a gint32 quantity to internal buffer.

Parameters

midi_file	the AgsMidiFile	
val	the value to write	

Since: 3.0.0

ags_midi_file_write_varlength ()

```
void
ags_midi_file_write_varlength (AgsMidiFile *midi_file,
                               long val Param2);
```

ags_midi_file_write_text ()

```
void
ags_midi_file_write_text (AgsMidiFile *midi_file,
                          gchar *text,
                          guint length);
```

Writes a string to internal buffer up to length bytes.

Parameters

midi_file	the AgsMidiFile	
text	the text	
length	the string's length	

Since: 3.0.0

ags_midi_file_read_header ()

```
unsigned char~*
ags_midi_file_read_header (AgsMidiFile *midi_file,
                           guint *buffer_length,
                           GError **error);
```

Reads the MIDI file's header and positions internal buffer pointer just behind it.

Parameters

midi_file	the AgsMidiFile	
buffer_length	pointer to return buffer length or NULL	
error	the GError pointer return location	

Returns

the header's bytes

Since: 3.0.0

ags_midi_file_write_header ()

```
void
ags_midi_file_write_header (AgsMidiFile *midi_file,
                            unsigned char *buffer,
                            guint buffer_length);
```

Write header bytes.

Parameters

midi_file	the AgsMidiFile	
buffer	the buffer to write	
buffer_length	the length of the buffer	

Since: 3.0.0

ags_midi_file_read_track_data ()

```
unsigned char~*
ags_midi_file_read_track_data (AgsMidiFile *midi_file,
                               guint *buffer_length,
                               GError **error);
```

Reads the MIDI file's track data.

Parameters

midi_file	the AgsMidiFile	
buffer_length	pointer to return buffer length or NULL	
error	the GError pointer return location	

Returns

the track's bytes

Since: 3.0.0

ags_midi_file_write_track_data ()

```
void
ags_midi_file_write_track_data (AgsMidiFile *midi_file,
                                unsigned char *buffer,
                                guint buffer_length);
```

ags_midi_file_read_notation ()

```
void
ags_midi_file_read_notation (AgsMidiFile *midi_file);
```

ags_mid_file_read_midi ()

```
void
ags_mid_file_read_midi (AgsMidiFile *midi_file);
```

ags_midi_file_new ()

```
AgsMidiFile~*
ags_midi_file_new ();
```

Create a new instance of AgsMidiFile

Returns

the new AgsMidiFile

Since: 3.0.0

AGS_IS_MIDI_FILE()

```
#define AGS_IS_MIDI_FILE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_MIDI_FILE))
```

AGS_IS_MIDI_FILE_CLASS()

```
#define AGS_IS_MIDI_FILE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_MIDI_FILE))
```

AGS_MIDI_FILE()

```
#define AGS_MIDI_FILE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MIDI_FILE, AgsMidiFile))
```

AGS_MIDI_FILE_CLASS()

```
#define AGS_MIDI_FILE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_MIDI_FILE, AgsMidiFileClass))
```

AGS_MIDI_FILE_GET_CLASS()

```
#define AGS_MIDI_FILE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_MIDI_FILE, AgsMidiFileClass))
```

ags_midi_file_get_type ()

```
GType
ags_midi_file_get_type (void);
```

Types and Values

AGS_MIDI_FILE_MAX_TEXT_LENGTH

```
#define AGS_MIDI_FILE_MAX_TEXT_LENGTH (4096)
```

AGS_MIDI_FILE_MTHD

```
#define AGS_MIDI_FILE_MTHD "MThd"
```

AGS_MIDI_FILE_MTRK

```
#define AGS_MIDI_FILE_MTRK "MTrk"
```

AGS_MIDI_FILE_DEFAULT_OFFSET

```
#define AGS_MIDI_FILE_DEFAULT_OFFSET (0)
```

AGS_MIDI_FILE_DEFAULT_FORMAT

```
#define AGS_MIDI_FILE_DEFAULT_FORMAT (1)
```

AGS_MIDI_FILE_DEFAULT_BEATS

```
#define AGS_MIDI_FILE_DEFAULT_BEATS (120)
```

AGS_MIDI_FILE_DEFAULT_FPS

```
#define AGS_MIDI_FILE_DEFAULT_FPS (30)
```

AGS_MIDI_FILE_DEFAULT_TICKS

```
#define AGS_MIDI_FILE_DEFAULT_TICKS (384)
```

enum AgsMidiFileStatus

Members

AGS_MIDI_FILE_KEY_OFF		
AGS_MIDI_FILE_KEY_ON		
AGS_MIDI_FILE_KEY_PRESSURE		
AGS_MIDI_FILE_CHANGE_PARAMETER		
AGS_MIDI_FILE_CHANGE_PROGRAM		
AGS_MIDI_FILE_CHANNEL_PRESSURE		

AGS_MIDI_FILE_CHANGE_PITCH_BEND		
AGS_MIDI_FILE_SYSEX		
AGS_MIDI_FILE_QUARTER_FRAME		
AGS_MIDI_FILE_SONG_POSITION		
AGS_MIDI_FILE_SONG_SELECT		
AGS_MIDI_FILE_UNDEFINED_0		
AGS_MIDI_FILE_UNDEFINED_1		
AGS_MIDI_FILE_TUNE_REQUEST		
AGS_MIDI_FILE_SYSEX_END		
AGS_MIDI_FILE_META_EVENT		

enum AgsMidiFileFlags

Members

AGS_MIDI_FILE_EOF		
AGS_MIDI_FILE_SMTPE		
AGS_MIDI_FILE_DROP_FRAME		

AGS_MIDI_FILE_ERROR

```
#define AGS_MIDI_FILE_ERROR (ags_midi_file_error_quark())
```

enum AgsMidiFileError

Members

AGS_MIDI_FILE_ERROR_PREMATURE_EOF		

struct AgsMidiFileTrack

```
struct AgsMidiFileTrack {
  GObject *sequencer;

  gchar *track_name;
  unsigned char *buffer;
};
```

AGS_TYPE_MIDI_FILE

```
#define AGS_TYPE_MIDI_FILE                (ags_midi_file_get_type ())
```

struct AgsMidiFile

```
struct AgsMidiFile;
```

struct AgsMidiFileClass

```
struct AgsMidiFileClass {
  GObjectClass gobject;
};
```

Property Details

The "filename" property

```
"filename"                  char~*
```

The assigned filename to perform input/output on.

Owner: AgsMidiFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

13.3 AgsMidiUtil

AgsMidiUtil — MIDI util

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_midi_util_is_key_on ()
gboolean	ags_midi_util_is_key_off ()
gboolean	ags_midi_util_is_key_pressure ()
gboolean	ags_midi_util_is_change_parameter ()
gboolean	ags_midi_util_is_pitch_bend ()
gboolean	ags_midi_util_is_change_program ()
gboolean	ags_midi_util_is_change_pressure ()
gboolean	ags_midi_util_is_sysex ()
gboolean	ags_midi_util_is_quarter_frame ()
gboolean	ags_midi_util_is_song_position ()
gboolean	ags_midi_util_is_song_select ()
gboolean	ags_midi_util_is_tune_request ()
gboolean	ags_midi_util_is_meta_event ()
gboolean	ags_midi_util_get_key_on ()
gboolean	ags_midi_util_get_key_off ()
gboolean	ags_midi_util_get_key_pressure ()
gboolean	ags_midi_util_get_change_parameter ()
gboolean	ags_midi_util_get_pitch_bend ()
gboolean	ags_midi_util_get_change_program ()
gboolean	ags_midi_util_get_change_pressure ()
gboolean	ags_midi_util_get_sysex ()
gboolean	ags_midi_util_get_quarter_frame ()
gboolean	ags_midi_util_get_song_position ()
gboolean	ags_midi_util_get_song_select ()
guchar *	ags_midi_util_to_smf ()
guint	ags_midi_util_delta_time_to_offset ()
glong	ags_midi_util_offset_to_delta_time ()
GType	ags_midi_util_get_type ()

Types and Values

#define	AGS_TYPE_MIDI_UTIL
struct	AgsMidiUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsMidiUtil
```

Includes

```
#include <ags/audio/midi/ags_midi_util.h>
```

Description

Utility functions for MIDI.

Functions

ags_midi_util_is_key_on ()

```
gboolean
ags_midi_util_is_key_on (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_key_off ()

```
gboolean
ags_midi_util_is_key_off (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_key_pressure ()

```
gboolean
ags_midi_util_is_key_pressure (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_change_parameter ()

```
gboolean
ags_midi_util_is_change_parameter (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_pitch_bend ()

```
gboolean
ags_midi_util_is_pitch_bend (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_change_program ()

```
gboolean
ags_midi_util_is_change_program (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_change_pressure ()

```
gboolean
ags_midi_util_is_change_pressure (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_sysex ()

```
gboolean
ags_midi_util_is_sysex (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_quarter_frame ()

```
gboolean
ags_midi_util_is_quarter_frame (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_song_position ()

```
gboolean
ags_midi_util_is_song_position (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_song_select ()

```
gboolean
ags_midi_util_is_song_select (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_tune_request ()

```
gboolean
ags_midi_util_is_tune_request (guchar *buffer);
```

Parameters

buffer	the midi buffer

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_is_meta_event ()

```
gboolean
ags_midi_util_is_meta_event (guchar *buffer);
```

Parameters

buffer	the midi buffer	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_midi_util_get_key_on ()

```
gboolean
ags_midi_util_get_key_on (guchar *buffer,
                          gint *channel,
                          gint *key,
                          gint *velocity);
```

Get key on fields of *buffer* .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
key	the return location of key.	*[out]*
velocity	the return location of velocity.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_key_off ()

```
gboolean
ags_midi_util_get_key_off (guchar *buffer,
                           gint *channel,
                           gint *key,
                           gint *velocity);
```

Get key off fields of *buffer* .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
key	the return location of key.	*[out]*
velocity	the return location of velocity.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_key_pressure ()

```
gboolean
ags_midi_util_get_key_pressure (guchar *buffer,
                                gint *channel,
                                gint *key,
                                gint *pressure);
```

Get key pressure fields of `buffer` .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
key	the return location of key.	*[out]*
pressure	the return location of pressure.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_change_parameter ()

```
gboolean
ags_midi_util_get_change_parameter (guchar *buffer,
                                    gint *channel,
                                    gint *control,
                                    gint *value);
```

Get change parameter fields of `buffer` .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
control	the return location of control.	*[out]*
value	the return location of value.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_pitch_bend ()

```
gboolean
ags_midi_util_get_pitch_bend (guchar *buffer,
                              gint *channel,
                              gint *pitch,
                              gint *transmitter);
```

Get pitch bend fields of *buffer* .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
pitch	the return location of pitch.	*[out]*
transmitter	the return location of transmitter.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_change_program ()

```
gboolean
ags_midi_util_get_change_program (guchar *buffer,
                                  gint *channel,
                                  gint *program);
```

Get change program fields of *buffer* .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
program	the return location of program.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_change_pressure ()

```
gboolean
ags_midi_util_get_change_pressure (guchar *buffer,
                                   gint *channel,
                                   gint *pressure);
```

Get change pressure fields of `buffer` .

Parameters

buffer	the MIDI buffer	
channel	the return location of channel.	*[out]*
pressure	the return location of pressure.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_sysex ()

```
gboolean
ags_midi_util_get_sysex (guchar *buffer,
                         guchar **data,
                         gint *length);
```

Get sysex fields of `buffer` .

Parameters

buffer	the MIDI buffer	
data	the return location of data.	*[out][transfer full]*
length	the return location of length.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_quarter_frame ()

```
gboolean
ags_midi_util_get_quarter_frame (guchar *buffer,
                                 gint *message_type,
                                 gint *values);
```

Get quarter frame fields of *buffer* .

Parameters

buffer	the MIDI buffer	
message_type	the return location of message type.	*[out]*
values	the return location of values.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_song_position ()

```
gboolean
ags_midi_util_get_song_position (guchar *buffer,
                                 gint *song_position);
```

Get song position fields of *buffer* .

Parameters

buffer	the MIDI buffer	
song_position	the return location of song position.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_get_song_select ()

```
gboolean
ags_midi_util_get_song_select (guchar *buffer,
                               gint *song_select);
```

Get song select fields of *buffer* .

Parameters

buffer	the MIDI buffer	
song_select	the return location of song select.	*[out]*

Returns

TRUE if successful, otherwise FALSE

Since: 3.13.0

ags_midi_util_to_smf ()

```
guchar~*
ags_midi_util_to_smf (guchar *midi_buffer,
                      guint buffer_length,
                      glong delta_time,
                      guint *smf_buffer_length);
```

Convert real-time MIDI to SMF.

Parameters

midi_buffer	the midi buffer	
buffer_length	the buffer length	
delta_time	the delta time	
smf_buffer_length	the return location of resulting length	

Returns

the SMF buffer

Since: 3.0.0

ags_midi_util_delta_time_to_offset ()

```
guint
ags_midi_util_delta_time_to_offset (gdouble delay_factor,
                                    glong division,
                                    glong tempo,
                                    glong bpm,
                                    glong delta_time);
```

Delta time to offset

Parameters

delay_factor	delay factor	
division	division	
tempo	tempo	
bpm	bpm	

delta_time	delta time	

Returns

the offset

Since: 3.0.0

ags_midi_util_offset_to_delta_time ()

```
glong
ags_midi_util_offset_to_delta_time (gdouble delay_factor,
                                    glong division,
                                    glong tempo,
                                    glong bpm,
                                    guint x);
```

Offset to delta time

Parameters

delay_factor	delay factor	
division	division	
tempo	tempo	
bpm	bpm	
x	note offset	

Returns

the delta time

Since: 3.0.0

ags_midi_util_get_type ()

```
GType
ags_midi_util_get_type (void);
```

Types and Values

AGS_TYPE_MIDI_UTIL

```
#define AGS_TYPE_MIDI_UTIL          (ags_midi_util_get_type())
```

struct AgsMidiUtil

```
struct AgsMidiUtil {
  //empty
};
```

13.4 AgsMidiBuilder

AgsMidiBuilder — MIDI builder

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MIDI_BUILDER_GET_OBJ_MUTEX()
#define	AGS_MIDI_BUILDER_HEADER()
#define	AGS_MIDI_BUILDER_TRACK()
AgsMidiBuilderHeader *	ags_midi_builder_header_alloc ()
void	ags_midi_builder_header_free ()
AgsMidiBuilderTrack *	ags_midi_builder_track_alloc ()
void	ags_midi_builder_track_free ()
GList *	ags_midi_builder_track_find_delta_time_with_track_name ()
void	ags_midi_builder_track_insert_midi_message ()
guchar *	ags_midi_builder_track_get_delta_time_offset ()
void	ags_midi_builder_midi_putc ()
void	ags_midi_builder_on_error ()
void	ags_midi_builder_append_header ()
void	ags_midi_builder_append_track ()
void	ags_midi_builder_append_key_on ()
void	ags_midi_builder_append_key_off ()
void	ags_midi_builder_append_key_pressure ()
void	ags_midi_builder_append_change_parameter ()
void	ags_midi_builder_append_change_pitch_bend ()
void	ags_midi_builder_append_change_program ()
void	ags_midi_builder_append_change_pressure ()
void	ags_midi_builder_append_sysex ()
void	ags_midi_builder_append_quarter_frame ()
void	ags_midi_builder_append_song_position ()
void	ags_midi_builder_append_song_select ()
void	ags_midi_builder_append_tune_request ()
void	ags_midi_builder_append_sequence_number ()
void	ags_midi_builder_append_smtpe ()
void	ags_midi_builder_append_tempo ()
void	ags_midi_builder_append_time_signature ()
void	ags_midi_builder_append_key_signature ()
void	ags_midi_builder_append_sequencer_meta_event ()
void	ags_midi_builder_append_text_event ()
void	ags_midi_builder_append_xml_node ()
void	ags_midi_builder_from_xml_doc ()
void	ags_midi_builder_build ()
void	ags_midi_builder_open_filename ()
guchar *	ags_midi_builder_get_data ()
guchar *	ags_midi_builder_get_data_with_length ()
void	ags_midi_builder_write ()
AgsMidiBuilder *	ags_midi_builder_new ()
AgsMidiBuilder *	ags_midi_builder_new_from_filename ()
#define	AGS_IS_MIDI_BUILDER()
#define	AGS_IS_MIDI_BUILDER_CLASS()

#define	AGS_MIDI_BUILDER()
#define	AGS_MIDI_BUILDER_CLASS()
#define	AGS_MIDI_BUILDER_GET_CLASS()
GType	ags_midi_builder_get_type ()

Properties

gpointer	file	Read / Write

Signals

void	append-header	Run Last
void	append-track	Run Last
void	change-channel-pressure	Run Last
void	change-parameter	Run Last
void	change-pitch-bend	Run Last
void	change-program	Run Last
void	key-off	Run Last
void	key-on	Run Last
void	key-pressure	Run Last
void	key-signature	Run Last
void	midi-putc	Run Last
void	on-error	Run Last
void	quarter-frame	Run Last
void	sequence-number	Run Last
void	sequencer-meta-event	Run Last
void	smtpe	Run Last
void	song-position	Run Last
void	song-select	Run Last
void	sysex	Run Last
void	tempo	Run Last
void	text-event	Run Last
void	time-signature	Run Last
void	tune-request	Run Last

Types and Values

enum	AgsMidiBuilderFlags
struct	AgsMidiBuilderHeader
struct	AgsMidiBuilderTrack
#define	AGS_TYPE_MIDI_BUILDER
struct	AgsMidiBuilder
struct	AgsMidiBuilderClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsMidiBuilder
```

Includes

```
#include <ags/audio/midi/ags_midi_builder.h>
```

Description

The AgsMidiBuilder lets you create MIDI files using its builder functions.

Functions

AGS_MIDI_BUILDER_GET_OBJ_MUTEX()

```
#define AGS_MIDI_BUILDER_GET_OBJ_MUTEX(obj) (&(((AgsMidiBuilder *) obj)->obj_mutex))
```

AGS_MIDI_BUILDER_HEADER()

```
#define AGS_MIDI_BUILDER_HEADER(ptr) ((AgsMidiBuilderHeader *)(ptr))
```

AGS_MIDI_BUILDER_TRACK()

```
#define AGS_MIDI_BUILDER_TRACK(ptr) ((AgsMidiBuilderTrack *)(ptr))
```

ags_midi_builder_header_alloc ()

```
AgsMidiBuilderHeader~*
ags_midi_builder_header_alloc ();
```

Allocate MIDI builder header.

Returns

the newly allocated AgsMidiBuilderHeader.

[type gpointer][transfer none]

Since: 3.0.0

ags_midi_builder_header_free ()

```
void
ags_midi_builder_header_free (AgsMidiBuilderHeader *midi_builder_header);
```

Free MIDI builder header.

Parameters

midi_builder_header	the AgsMidiBuilderHeader.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_midi_builder_track_alloc ()

```
AgsMidiBuilderTrack~*
ags_midi_builder_track_alloc ();
```

Allocate MIDI builder track.

Returns

the newly allocated AgsMidiBuilderTrack.

[type gpointer][transfer none]

Since: 3.0.0

ags_midi_builder_track_free ()

```
void
ags_midi_builder_track_free (AgsMidiBuilderTrack *midi_builder_track);
```

Free MIDI builder track.

Parameters

midi_builder_track	the AgsMidiBuilderTrack.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_midi_builder_track_find_delta_time_with_track_name ()

```
GList~*
ags_midi_builder_track_find_delta_time_with_track_name
                                (GList *midi_builder_track,
                                 guint64 absolute_time,
                                 gchar *track_name);
```

Finds matching AgsMidiBuilderTrack within `midi_builder_track` GList.

Parameters

midi_builder_track	the AgsMidiBuilderTrack.	*[element-type gpointer][transfer none]*
absolute_time	the absolute time	
track_name	the track's string representation	

Returns

the matching GList.

[element-type gpointer][transfer none]

Since: 3.0.0

ags_midi_builder_track_insert_midi_message ()

```
void
ags_midi_builder_track_insert_midi_message
                                (AgsMidiBuilderTrack *midi_builder_track,
                                 guchar *buffer,
                                 guint length);
```

Insert MIDI message.

Parameters

midi_builder_track	the AgsMidiBuilderTrack	
buffer	the MIDI message	
length	the buffer length	

Since: 3.0.0

ags_midi_builder_track_get_delta_time_offset ()

```
guchar~*
ags_midi_builder_track_get_delta_time_offset
                                (AgsMidiBuilderTrack *midi_builder_track,
                                 guint64 absolute_time);
```

Get offset by delta time.

Parameters

midi_builder_track	the AgsMidiBuilderTrack	
absolute_time	the absolute time	

Returns

the buffer position before *delta_time*

Since: 3.0.0

ags_midi_builder_midi_putc ()

```
void
ags_midi_builder_midi_putc (AgsMidiBuilder *midi_builder,
                            gint c);
```

Put char in MIDI file.

Parameters

midi_builder	the AgsMidiBuilder	
c	the character to put	

Since: 3.0.0

ags_midi_builder_on_error ()

```
void
ags_midi_builder_on_error (AgsMidiBuilder *midi_builder,
                           GError **error);
```

Triggered as an error occurs.

Parameters

midi_builder	the AgsMidiBuilder	
error	the error	

Since: 3.0.0

ags_midi_builder_append_header ()

```
void
ags_midi_builder_append_header (AgsMidiBuilder *midi_builder,
                                guint offset,
                                guint format,
                                guint track_count,
                                guint division,
                                guint times,
                                guint bpm,
                                guint clicks);
```

Appends MIDI header to *midi_builder* .

Parameters

midi_builder	the AgsMidiBuilder	
offset	start delta-time	
format	either 0, 1 or 2.	
track_count	the number of tracks	
division	timing division	
times	pulse per quarter note	
bpm	beats per minute	
clicks	timing clicks	

Since: 3.0.0

ags_midi_builder_append_track ()

```
void
ags_midi_builder_append_track (AgsMidiBuilder *midi_builder,
                               gchar *track_name);
```

Append a track called *track_name* to *midi_builder* .

Parameters

midi_builder	the AgsMidiBuilder	
track_name	the track name	

Since: 3.0.0

ags_midi_builder_append_key_on ()

```
void
ags_midi_builder_append_key_on (AgsMidiBuilder *midi_builder,
                                guint delta_time,
                                guint audio_channel,
                                guint note,
                                guint velocity);
```

Append key-on for *note* to *midi_builder* with key dynamics *velocity* , at *delta_time* .

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	delta-time	
audio_channel	the audio channel	
note	the note to play from 0 to 128	
velocity	key dynamics	

Since: 3.0.0

ags_midi_builder_append_key_off ()

```
void
ags_midi_builder_append_key_off (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint audio_channel,
                                 guint note,
                                 guint velocity);
```

Append key-off for *note* to *midi_builder* with key dynamics *velocity* , at *delta_time* .

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	delta-time	
audio_channel	the audio channel	
note	the note to play from 0 to 128	
velocity	key dynamics	

Since: 3.0.0

ags_midi_builder_append_key_pressure ()

```
void
ags_midi_builder_append_key_pressure (AgsMidiBuilder *midi_builder,
                                      guint delta_time,
                                      guint audio_channel,
                                      guint note,
                                      guint pressure);
```

Append key-pressure for *note* to *midi_builder* with key dynamics *pressure* , at *delta_time* .

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	delta-time	
audio_channel	the audio channel	
note	the note to play from 0 to 128	
pressure	key dynamics	

Since: 3.0.0

ags_midi_builder_append_change_parameter ()

```
void
ags_midi_builder_append_change_parameter
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint channel,
                                 guint control,
                                 guint value);
```

Appends change parameter.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
channel	the channel	
control	the control	
value	the value	

Since: 3.0.0

ags_midi_builder_append_change_pitch_bend ()

```
void
ags_midi_builder_append_change_pitch_bend
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint channel,
                                 guint pitch,
                                 guint transmitter);
```

Change pitch bend.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
channel	the channel	
pitch	the pitch	
transmitter	the transmitter	

Since: 3.0.0

ags_midi_builder_append_change_program ()

```
void
ags_midi_builder_append_change_program
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint channel,
                                 guint program);
```

Appends change program.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
channel	the channel	
program	the program	

Since: 3.0.0

ags_midi_builder_append_change_pressure ()

```
void
ags_midi_builder_append_change_pressure
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint channel,
                                 guint pressure);
```

Appends change pressure.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
channel	the channel	
pressure	the pressure	

Since: 3.0.0

ags_midi_builder_append_sysex ()

```
void
ags_midi_builder_append_sysex (AgsMidiBuilder *midi_builder,
                               guint delta_time,
                               guchar *sysex_data,
                               guint length);
```

Appends sysex data.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
sysex_data	the sysex data	
length	length	

Since: 3.0.0

ags_midi_builder_append_quarter_frame ()

```
void
ags_midi_builder_append_quarter_frame (AgsMidiBuilder *midi_builder,
                                       guint delta_time,
                                       guint message_type,
                                       guint values);
```

Appends quarter frame.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
message_type	the message type	
values	values	

Since: 3.0.0

ags_midi_builder_append_song_position ()

```
void
ags_midi_builder_append_song_position (AgsMidiBuilder *midi_builder,
                                       guint delta_time,
                                       guint song_position);
```

Appends song position.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
song_position	the song position	

Since: 3.0.0

ags_midi_builder_append_song_select ()

```
void
ags_midi_builder_append_song_select (AgsMidiBuilder *midi_builder,
                                     guint delta_time,
                                     guint song_select);
```

Appends song select.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
song_select	the song select	

Since: 3.0.0

ags_midi_builder_append_tune_request ()

```
void
ags_midi_builder_append_tune_request (AgsMidiBuilder *midi_builder,
                                      guint delta_time);
```

Appends tune request.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	

Since: 3.0.0

ags_midi_builder_append_sequence_number ()

```
void
ags_midi_builder_append_sequence_number
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint sequence);
```

Appends sequence number.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
sequence	the sequence number	

Since: 3.0.0

ags_midi_builder_append_smtpe ()

```
void
ags_midi_builder_append_smtpe (AgsMidiBuilder *midi_builder,
                               guint delta_time,
                               guint rr,
                               guint hr,
                               guint mn,
                               guint se,
                               guint fr);
```

Appends smtpe.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
rr	rate	
hr	hours	
mn	minutes	
se	seconds	
fr	fraction	

Since: 3.0.0

ags_midi_builder_append_tempo ()

```
void
ags_midi_builder_append_tempo (AgsMidiBuilder *midi_builder,
                               guint delta_time,
                               guint tempo);
```

Appends tempo.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
tempo	the tempo	

Since: 3.0.0

ags_midi_builder_append_time_signature ()

```
void
ags_midi_builder_append_time_signature
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint nn,
                                 guint dd,
                                 guint cc,
                                 guint bb);
```

Appends time signature.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
nn	numerator	
dd	denominator	
cc	ticks per metronome click	
bb	32nd per quarter note	

Since: 3.0.0

ags_midi_builder_append_key_signature ()

```
void
ags_midi_builder_append_key_signature (AgsMidiBuilder *midi_builder,
                                       guint delta_time,
                                       guint sf,
                                       guint mi);
```

Appends key signature.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
sf	key	
mi	mode	

Since: 3.0.0

ags_midi_builder_append_sequencer_meta_event ()

```
void
ags_midi_builder_append_sequencer_meta_event
                                (AgsMidiBuilder *midi_builder,
                                 guint delta_time,
                                 guint len,
                                 guint id,
                                 guint data);
```

Appends sequencer meta event.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
len	the length	
id	the id	
data	the data	

Since: 3.0.0

ags_midi_builder_append_text_event ()

```
void
ags_midi_builder_append_text_event (AgsMidiBuilder *midi_builder,
                                    guint delta_time,
                                    gchar *text,
                                    guint length);
```

Appends text event.

Parameters

midi_builder	the AgsMidiBuilder	
delta_time	the delta time	
text	the string	
length	the string length	

Since: 3.0.0

ags_midi_builder_append_xml_node ()

```
void
ags_midi_builder_append_xml_node (AgsMidiBuilder *midi_builder,
                                  xmlNode *node);
```

Append from XML node *node* .

Parameters

midi_builder	the AgsMidiBuilder	
node	the xmlNode	

Since: 3.0.0

ags_midi_builder_from_xml_doc ()

```
void
ags_midi_builder_from_xml_doc (AgsMidiBuilder *midi_builder,
                               xmlDoc *doc);
```

Builds from XML document *doc* .

Parameters

midi_builder	the AgsMidiBuilder	
doc	the xmlDocument	

Since: 3.0.0

ags_midi_builder_build ()

```
void
ags_midi_builder_build (AgsMidiBuilder *midi_builder);
```

Build the MIDI data.

Parameters

midi_builder	the AgsMidiBuilder	

Since: 3.0.0

ags_midi_builder_open_filename ()

```
void
ags_midi_builder_open_filename (AgsMidiBuilder *midi_builder,
                                gchar *filename);
```

Open *filename* and assign FILE stream to *midi_builder* .

Parameters

midi_builder	the AgsMidiBuilder	
filename	the filename	

Since: 3.2.0

ags_midi_builder_get_data ()

```
guchar~*
ags_midi_builder_get_data (AgsMidiBuilder *midi_builder);
```

Get SMF data of *midi_builder* .

Parameters

midi_builder	the AgsMidiBuilder	

Returns

the SMF data

Since: 3.2.0

ags_midi_builder_get_data_with_length ()

```
guchar~*
ags_midi_builder_get_data_with_length (AgsMidiBuilder *midi_builder,
                                       guint *length);
```

Get SMF data of *midi_builder* .

Parameters

midi_builder	the AgsMidiBuilder	
length	the length of data.	*[out]*

Returns

the MIDI data as array.

[transfer full]

Since: 3.7.24

ags_midi_builder_write ()

```
void
ags_midi_builder_write (AgsMidiBuilder *midi_builder);
```

Write data of *midi_builder* to file.

Parameters

midi_builder	the AgsMidiBuilder	

Since: 3.2.0

ags_midi_builder_new ()

```
AgsMidiBuilder~*
ags_midi_builder_new ();
```

Creates a new instance of AgsMidiBuilder

Returns

the new AgsMidiBuilder

Since: 3.0.0

ags_midi_builder_new_from_filename ()

```
AgsMidiBuilder~*
ags_midi_builder_new_from_filename (gchar *filename);
```

Creates a new instance of AgsMidiBuilder

Parameters

filename	the filename.	*[nullable]*

Returns

the new AgsMidiBuilder

Since: 3.2.0

AGS_IS_MIDI_BUILDER()

```
#define AGS_IS_MIDI_BUILDER(obj)            (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_MIDI_BUILDER))
```

AGS_IS_MIDI_BUILDER_CLASS()

```
#define AGS_IS_MIDI_BUILDER_CLASS(class)    (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_MIDI_BUILDER))
```

AGS_MIDI_BUILDER()

```
#define AGS_MIDI_BUILDER(obj)               (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MIDI_BUILDER, AgsMidiBuilder))
```

AGS_MIDI_BUILDER_CLASS()

```
#define AGS_MIDI_BUILDER_CLASS(class)       (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_MIDI_BUILDER, AgsMidiBuilderClass))
```

AGS_MIDI_BUILDER_GET_CLASS()

```
#define AGS_MIDI_BUILDER_GET_CLASS(obj)     (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_MIDI_BUILDER, AgsMidiBuilderClass))
```

ags_midi_builder_get_type ()

```
GType
ags_midi_builder_get_type (void);
```

Types and Values

enum AgsMidiBuilderFlags

Members

AGS_MIDI_BUILDER_EOF	
AGS_MIDI_BUILDER_EOT	

struct AgsMidiBuilderHeader

```
struct AgsMidiBuilderHeader {
  guint offset;
  guint format;
  guint count;
  guint division;
  guint times;
  guint beat;
  guint clicks;

  guchar *data;
  guint length;
};
```

struct AgsMidiBuilderTrack

```
struct AgsMidiBuilderTrack {
  AgsMidiBuilder *midi_builder;

  guint offset;
  gchar *track_name;

  guint64 absolute_time;

  guchar *data;
  guint length;
};
```

AGS_TYPE_MIDI_BUILDER

```
#define AGS_TYPE_MIDI_BUILDER                (ags_midi_builder_get_type ())
```

struct AgsMidiBuilder

```
struct AgsMidiBuilder;
```

struct AgsMidiBuilderClass

```
struct AgsMidiBuilderClass {
  GObjectClass gobject;

  void (*midi_putc)(AgsMidiBuilder *midi_builder,
        gint c);
  void (*on_error)(AgsMidiBuilder *midi_builder,
       GError **error);

  /* channel messages */
  void (*append_header)(AgsMidiBuilder *midi_builder,
      guint offset, guint format,
```

```
    guint track_count, guint division,
    guint times, guint bpm,
    guint clicks);
void (*append_track)(AgsMidiBuilder *midi_builder,
         gchar *track_name);

void (*append_key_on)(AgsMidiBuilder *midi_builder,
    guint delta_time,
    guint audio_channel,
    guint note,
    guint velocity);
void (*append_key_off)(AgsMidiBuilder *midi_builder,
     guint delta_time,
     guint audio_channel,
     guint note,
     guint velocity);
void (*append_key_pressure)(AgsMidiBuilder *midi_builder,
          guint delta_time,
          guint audio_channel,
          guint note,
          guint pressure);

void (*append_change_parameter)(AgsMidiBuilder *midi_builder,
        guint delta_time,
        guint channel,
        guint control,
        guint value);
void (*append_change_pitch_bend)(AgsMidiBuilder *midi_builder,
         guint delta_time,
         guint channel,
         guint pitch,
         guint transmitter);
void (*append_change_program)(AgsMidiBuilder *midi_builder,
      guint delta_time,
      guint channel,
      guint program);
void (*append_change_pressure)(AgsMidiBuilder *midi_builder,
       guint delta_time,
       guint channel,
       guint pressure);

/* sysex and system common */
void (*append_sysex)(AgsMidiBuilder *midi_builder,
         guint delta_time,
         guchar *sysex_data, guint length);

void (*append_quarter_frame)(AgsMidiBuilder *midi_builder,
           guint delta_time,
           guint message_type,
           guint values);
void (*append_song_position)(AgsMidiBuilder *midi_builder,
           guint delta_time,
           guint song_position);
void (*append_song_select)(AgsMidiBuilder *midi_builder,
         guint delta_time,
         guint song_select);
void (*append_tune_request)(AgsMidiBuilder *midi_builder,
       guint delta_time);

/* meta events */
void (*append_sequence_number)(AgsMidiBuilder *midi_builder,
       guint delta_time,
```

```
        guint sequence);
  void (*append_smtpe)(AgsMidiBuilder *midi_builder,
          guint delta_time,
          guint rr, guint hr, guint mn, guint se, guint fr);
  void (*append_tempo)(AgsMidiBuilder *midi_builder,
          guint delta_time,
          guint tempo);

  void (*append_time_signature)(AgsMidiBuilder *midi_builder,
        guint delta_time,
        guint nn, guint dd, guint cc, guint bb);
  void (*append_key_signature)(AgsMidiBuilder *midi_builder,
            guint delta_time,
            guint sf, guint mi);
  void (*append_sequencer_meta_event)(AgsMidiBuilder *midi_builder,
             guint delta_time,
             guint len, guint id, guint data);
  void (*append_text_event)(AgsMidiBuilder *midi_builder,
         guint delta_time,
         gchar *text, guint length);
};
```

Property Details

The "file" property

```
  "file"                     gpointer
```

The file to parse data from.

Owner: AgsMidiBuilder

Flags: Read / Write

Since: 3.0.0

Signal Details

The "append-header" signal

```
void
user_function (AgsMidiBuilder *midi_builder,
               guint           offset,
               guint           format,
               guint           track_count,
               guint           division,
               guint           times,
               guint           bpm,
               guint           clicks,
               gpointer        user_data)
```

The ::append-header signal is emited during building of header.

Parameters

midi_builder	the builder	

offset	start delta-time	
format	format 0, 1 or 2	
track_count	the number of tracks	
division	timing division	
times	pulses per quarter note	
bpm	beats per minute	
clicks	timing clicks	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "append-track" signal

```
void
user_function (AgsMidiBuilder *midi_builder,
               char           *track_name,
               gpointer        user_data)
```

The ::append-track signal is emited during building of track.

Parameters

midi_builder	the builder	
track_name	the trach name	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "change-channel-pressure" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               gpointer        user_data)
```

Flags: Run Last

The "change-parameter" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               gpointer        user_data)
```

Flags: Run Last

The "change-pitch-bend" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               gpointer        user_data)
```

Flags: Run Last

The "change-program" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               gpointer        user_data)
```

Flags: Run Last

The "key-off" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               gpointer        user_data)
```

Flags: Run Last

The "key-on" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               gpointer        user_data)
```

Flags: Run Last

The "key-pressure" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               gpointer        user_data)
```

Flags: Run Last

The "key-signature" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               gboolean        arg3,
               gpointer        user_data)
```

Flags: Run Last

The "midi-putc" signal

```
void
user_function (AgsMidiBuilder *midi_builder,
               int             error,
               gpointer        user_data)
```

The ::midi-putc signal is emited during putting char to file.

Parameters

midi_builder	the builder	
error	the GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "on-error" signal

```
void
user_function (AgsMidiBuilder *midi_builder,
               gpointer        error,
               gpointer        user_data)
```

The ::on-error signal is emited during building of event.

Parameters

midi_builder	the builder	
error	the GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "quarter-frame" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               gpointer        user_data)
```

Flags: Run Last

The "sequence-number" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               gpointer        user_data)
```

Flags: Run Last

The "sequencer-meta-event" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               gpointer        user_data)
```

Flags: Run Last

The "smtpe" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               guint           arg5,
               guint           arg6,
               gpointer        user_data)
```

Flags: Run Last

The "song-position" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               gpointer        user_data)
```

Flags: Run Last

The "song-select" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               gpointer        user_data)
```

Flags: Run Last

The "sysex" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               gpointer        arg2,
               guint           arg3,
               gpointer        user_data)
```

Flags: Run Last

The "tempo" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               gpointer        user_data)
```

Flags: Run Last

The "text-event" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               char           *arg2,
               guint           arg3,
               gpointer        user_data)
```

Flags: Run Last

The "time-signature" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               guint           arg2,
               guint           arg3,
               guint           arg4,
               guint           arg5,
               gpointer        user_data)
```

Flags: Run Last

The "tune-request" signal

```
void
user_function (AgsMidiBuilder *agsmidibuilder,
               guint           arg1,
               gpointer        user_data)
```

Flags: Run Last

13.5 AgsMidiParser

AgsMidiParser — the MIDI parser

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MIDI_PARSER_GET_OBJ_MUTEX()
gint16	ags_midi_parser_read_gint16 ()
gint32	ags_midi_parser_read_gint24 ()
gint32	ags_midi_parser_read_gint32 ()
long	ags_midi_parser_read_varlength ()
gchar *	ags_midi_parser_read_text ()
gdouble	ags_midi_parser_ticks_to_sec ()
int	ags_midi_parser_midi_getc ()
void	ags_midi_parser_on_error ()
xmlDoc *	ags_midi_parser_parse_full ()
xmlNode *	ags_midi_parser_parse_bytes ()
xmlNode *	ags_midi_parser_parse_header ()
xmlNode *	ags_midi_parser_parse_track ()
xmlNode *	ags_midi_parser_channel_message ()
xmlNode *	ags_midi_parser_key_on ()
xmlNode *	ags_midi_parser_key_off ()
xmlNode *	ags_midi_parser_key_pressure ()
xmlNode *	ags_midi_parser_change_parameter ()
xmlNode *	ags_midi_parser_change_pitch_bend ()
xmlNode *	ags_midi_parser_change_program ()
xmlNode *	ags_midi_parser_change_channel_pressure ()
xmlNode *	ags_midi_parser_not_defined ()
xmlNode *	ags_midi_parser_sysex ()
xmlNode *	ags_midi_parser_system_common ()
xmlNode *	ags_midi_parser_meta_event ()
xmlNode *	ags_midi_parser_sequence_number ()
xmlNode *	ags_midi_parser_end_of_track ()
xmlNode *	ags_midi_parser_smtpe ()
xmlNode *	ags_midi_parser_tempo ()
xmlNode *	ags_midi_parser_time_signature ()
xmlNode *	ags_midi_parser_key_signature ()
xmlNode *	ags_midi_parser_sequencer_meta_event ()
xmlNode *	ags_midi_parser_meta_misc ()
xmlNode *	ags_midi_parser_text_event ()

void	ags_midi_parser_open_filename ()
void	ags_midi_parser_set_buffer ()
void	ags_midi_parser_set_file_length ()
AgsMidiParser *	ags_midi_parser_new ()
AgsMidiParser *	ags_midi_parser_new_from_filename ()
#define	AGS_IS_MIDI_PARSER()
#define	AGS_IS_MIDI_PARSER_CLASS()
#define	AGS_MIDI_PARSER()
#define	AGS_MIDI_PARSER_CLASS()
#define	AGS_MIDI_PARSER_GET_CLASS()
GType	ags_midi_parser_get_type ()

Properties

gpointer	file	Read / Write

Signals

gpointer	change-channel-pressure	Run Last
gpointer	change-parameter	Run Last
gpointer	change-pitch-bend	Run Last
gpointer	change-program	Run Last
gpointer	end-of-track	Run Last
gpointer	key-off	Run Last
gpointer	key-on	Run Last
gpointer	key-pressure	Run Last
gpointer	key-signature	Run Last
gpointer	meta-event	Run Last
int	midi-getc	Run Last
gpointer	not-defined	Run Last
void	on-error	Run Last
gpointer	parse-bytes	Run Last
gpointer	parse-full	Run Last
gpointer	parse-header	Run Last
gpointer	parse-track	Run Last
gpointer	sequence-number	Run Last
gpointer	sequencer-meta-event	Run Last
gpointer	smtpe	Run Last
gpointer	sysex	Run Last
gpointer	system-common	Run Last
gpointer	tempo	Run Last
gpointer	text-event	Run Last
gpointer	time-signature	Run Last

Types and Values

#define	AGS_MIDI_PARSER_MAX_TEXT_LENGTH
#define	AGS_MIDI_PARSER_MTHD
#define	AGS_MIDI_PARSER_MTCK
#define	AGS_MIDI_EVENT
enum	AgsMidiParserFlags
enum	AgsMidiChunkFlags

#define	AGS_TYPE_MIDI_PARSER
struct	AgsMidiParser
struct	AgsMidiParserClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsMidiParser
```

Includes

```
#include <ags/audio/midi/ags_midi_parser.h>
```

Description

AgsMidiParser parses your midi.

Functions

AGS_MIDI_PARSER_GET_OBJ_MUTEX()

```
#define AGS_MIDI_PARSER_GET_OBJ_MUTEX(obj) (&(((AgsMidiParser *) obj)->obj_mutex))
```

ags_midi_parser_read_gint16 ()

```
gint16
ags_midi_parser_read_gint16 (AgsMidiParser *midi_parser);
```

Read gint16.

Parameters

midi_parser	the AgsMidiParser

Returns

the read gint16

Since: 3.0.0

ags_midi_parser_read_gint24 ()

```
gint32
ags_midi_parser_read_gint24 (AgsMidiParser *midi_parser);
```

Read gint32.

Parameters

midi_parser	the AgsMidiParser

Returns

the read gint32

Since: 3.0.0

ags_midi_parser_read_gint32 ()

```
gint32
ags_midi_parser_read_gint32 (AgsMidiParser *midi_parser);
```

Read gint32.

Parameters

midi_parser	the AgsMidiParser

Returns

the read gint32

Since: 3.0.0

ags_midi_parser_read_varlength ()

```
long
ags_midi_parser_read_varlength (AgsMidiParser *midi_parser);
```

Read varlength as long.

Parameters

midi_parser	the AgsMidiParser

Returns

the read varlength

Since: 3.0.0

ags_midi_parser_read_text ()

```
gchar~*
ags_midi_parser_read_text (AgsMidiParser *midi_parser,
                           gint length);
```

Read text.

Parameters

midi_parser	the AgsMidiParser	
length	the length	

Returns

the text read as string

Since: 3.0.0

ags_midi_parser_ticks_to_sec ()

```
gdouble
ags_midi_parser_ticks_to_sec (AgsMidiParser *midi_parser,
                              guint ticks,
                              gint division,
                              guint tempo);
```

Convert ticks to seconds.

Parameters

midi_parser	the AgsMidiParser	
ticks	the ticks count	
division	the division	
tempo	the tempo	

Returns

the seconds read from ticks

Since: 3.0.0

ags_midi_parser_midi_getc ()

```
int
ags_midi_parser_midi_getc (AgsMidiParser *midi_parser);
```

Read one byte.

Parameters

midi_parser	the AgsMidiParser	

Returns

the byte read

Since: 3.0.0

ags_midi_parser_on_error ()

```
void
ags_midi_parser_on_error (AgsMidiParser *midi_parser,
                          GError **error);
```

Error reporting.

Parameters

midi_parser	the AgsMidiParser	
error	the return location of GError	

Since: 3.0.0

ags_midi_parser_parse_full ()

```
xmlDoc~*
ags_midi_parser_parse_full (AgsMidiParser *midi_parser);
```

Parse all data of "file".

Parameters

midi_parser	the AgsMidiParser	

Returns

the xmlDoc representing your MIDI file.

[transfer none]

Since: 3.0.0

ags_midi_parser_parse_bytes ()

```
xmlNode~*
ags_midi_parser_parse_bytes (AgsMidiParser *midi_parser,
                             guchar *midi_buffer,
                             guint buffer_length);
```

ags_midi_parser_parse_header ()

```
xmlNode~*
ags_midi_parser_parse_header (AgsMidiParser *midi_parser);
```

Parse MIDI header

Parameters

midi_parser	the AgsMidiParser

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_parse_track ()

```
xmlNode~*
ags_midi_parser_parse_track (AgsMidiParser *midi_parser);
```

Parse MIDI track.

Parameters

midi_parser	the AgsMidiParser

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_channel_message ()

```
xmlNode~*
ags_midi_parser_channel_message (AgsMidiParser *midi_parser,
                                 guint status);
```

Parse channel message.

Parameters

midi_parser	the AgsMidiParser
status	the status byte

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_key_on ()

```
xmlNode~*
ags_midi_parser_key_on (AgsMidiParser *midi_parser,
                        guint status);
```

Parse key on.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_key_off ()

```
xmlNode~*
ags_midi_parser_key_off (AgsMidiParser *midi_parser,
                         guint status);
```

Parse key off.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_key_pressure ()

```
xmlNode~*
ags_midi_parser_key_pressure (AgsMidiParser *midi_parser,
                              guint status);
```

Parse key pressure.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_change_parameter ()

```
xmlNode~*
ags_midi_parser_change_parameter (AgsMidiParser *midi_parser,
                                  guint status);
```

Parse change parameter.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_change_pitch_bend ()

```
xmlNode~*
ags_midi_parser_change_pitch_bend (AgsMidiParser *midi_parser,
                                   guint status);
```

Parse change pitch bend.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_change_program ()

```
xmlNode~*
ags_midi_parser_change_program (AgsMidiParser *midi_parser,
                                guint status);
```

Parse change program.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_change_channel_pressure ()

```
xmlNode~*
ags_midi_parser_change_channel_pressure
                                (AgsMidiParser *midi_parser,
                                 guint status);
```

Parse change channel pressure.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_not_defined ()

```
xmlNode~*
ags_midi_parser_not_defined (AgsMidiParser *midi_parser,
                             guint status);
```

Parse not defined.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_sysex ()

```
xmlNode~*
ags_midi_parser_sysex (AgsMidiParser *midi_parser,
                       guint status);
```

Parse sysex.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_system_common ()

```
xmlNode~*
ags_midi_parser_system_common (AgsMidiParser *midi_parser,
                               guint status);
```

Parse system common.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_meta_event ()

```
xmlNode~*
ags_midi_parser_meta_event (AgsMidiParser *midi_parser,
                            guint status);
```

Parse meta event.

Parameters

midi_parser	the AgsMidiParser	
status	the status byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_sequence_number ()

```
xmlNode~*
ags_midi_parser_sequence_number (AgsMidiParser *midi_parser,
                                 guint meta_type);
```

Parse sequence number.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_end_of_track ()

```
xmlNode~*
ags_midi_parser_end_of_track (AgsMidiParser *midi_parser,
                              guint meta_type);
```

Parse end of track.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_smtpe ()

```
xmlNode~*
ags_midi_parser_smtpe (AgsMidiParser *midi_parser,
                       guint meta_type);
```

Parse smtpe.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_tempo ()

```
xmlNode~*
ags_midi_parser_tempo (AgsMidiParser *midi_parser,
                       guint meta_type);
```

Parse tempo.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_time_signature ()

```
xmlNode~*
ags_midi_parser_time_signature (AgsMidiParser *midi_parser,
                                guint meta_type);
```

Parse time signature.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_key_signature ()

```
xmlNode~*
ags_midi_parser_key_signature (AgsMidiParser *midi_parser,
                               guint meta_type);
```

Parse key signature.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_sequencer_meta_event ()

```
xmlNode~*
ags_midi_parser_sequencer_meta_event (AgsMidiParser *midi_parser,
                                      guint meta_type);
```

ags_midi_parser_meta_misc ()

```
xmlNode~*
ags_midi_parser_meta_misc (AgsMidiParser *midi_parser,
                           guint meta_type);
```

ags_midi_parser_text_event ()

```
xmlNode~*
ags_midi_parser_text_event (AgsMidiParser *midi_parser,
                            guint meta_type);
```

Parse text event.

Parameters

midi_parser	the AgsMidiParser	
meta_type	the meta type byte	

Returns

the xmlNode containing parsed data.

[transfer none]

Since: 3.0.0

ags_midi_parser_open_filename ()

```
void
ags_midi_parser_open_filename (AgsMidiParser *midi_parser,
                               gchar *filename);
```

Open *filename* and assign FILE stream to *midi_parser* .

Parameters

midi_parser	the AgsMidiParser	
filename	the filename	

Since: 3.2.0

ags_midi_parser_set_buffer ()

```
void
ags_midi_parser_set_buffer (AgsMidiParser *midi_parser,
                            guchar *buffer);
```

Set buffer of *midi_parser*

Parameters

midi_parser	the AgsMidiParser	
buffer	the buffer	

Since: 3.2.0

ags_midi_parser_set_file_length ()

```
void
ags_midi_parser_set_file_length (AgsMidiParser *midi_parser,
                                 size_t file_length);
```

Set file length of *midi_parser*

Parameters

midi_parser	the AgsMidiParser	
file_length	the file length	

Since: 3.7.38

ags_midi_parser_new ()

```
AgsMidiParser~*
ags_midi_parser_new ();
```

Creates a new instance of AgsMidiParser

Returns

the new AgsMidiParser

Since: 3.0.0

ags_midi_parser_new_from_filename ()

```
AgsMidiParser~*
ags_midi_parser_new_from_filename (gchar *filename);
```

Creates a new instance of AgsMidiParser

Parameters

filename	the filename.	*[nullable]*

Returns

the new AgsMidiParser

Since: 3.2.0

AGS_IS_MIDI_PARSER()

```
#define AGS_IS_MIDI_PARSER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_MIDI_PARSER))
```

AGS_IS_MIDI_PARSER_CLASS()

```
#define AGS_IS_MIDI_PARSER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_MIDI_PARSER))
```

AGS_MIDI_PARSER()

```
#define AGS_MIDI_PARSER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MIDI_PARSER, AgsMidiParser))
```

AGS_MIDI_PARSER_CLASS()

```
#define AGS_MIDI_PARSER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_MIDI_PARSER, AgsMidiParserClass))
```

AGS_MIDI_PARSER_GET_CLASS()

```
#define AGS_MIDI_PARSER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_MIDI_PARSER, AgsMidiParserClass))
```

ags_midi_parser_get_type ()

```
GType
ags_midi_parser_get_type (void);
```

Types and Values

AGS_MIDI_PARSER_MAX_TEXT_LENGTH

```
#define AGS_MIDI_PARSER_MAX_TEXT_LENGTH (4096)
```

AGS_MIDI_PARSER_MTHD

```
#define AGS_MIDI_PARSER_MTHD "MThd"
```

AGS_MIDI_PARSER_MTCK

```
#define AGS_MIDI_PARSER_MTCK "MTrk"
```

AGS_MIDI_EVENT

```
#define AGS_MIDI_EVENT "event"
```

enum AgsMidiParserFlags

Members

AGS_MIDI_PARSER_EOF		
AGS_MIDI_PARSER_EOT		

enum AgsMidiChunkFlags

Members

AGS_MIDI_CHUNK_HEADER		
AGS_MIDI_CHUNK_TRACK		
AGS_MIDI_CHUNK_UNKNOWN		

AGS_TYPE_MIDI_PARSER

```
#define AGS_TYPE_MIDI_PARSER                (ags_midi_parser_get_type ())
```

struct AgsMidiParser

```
struct AgsMidiParser;
```

struct AgsMidiParserClass

```
struct AgsMidiParserClass {
  GObjectClass gobject;

  int (*midi_getc)(AgsMidiParser *midi_parser);
  void (*on_error)(AgsMidiParser *midi_parser,
       GError **error);

  xmlDoc* (*parse_full)(AgsMidiParser *midi_parser);
  xmlNode* (*parse_bytes)(AgsMidiParser *midi_parser,
        guchar *midi_buffer,
        guint buffer_length);

  xmlNode* (*parse_header)(AgsMidiParser *midi_parser);
  xmlNode* (*parse_track)(AgsMidiParser *midi_parser);

  xmlNode* (*key_on)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*key_off)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*key_pressure)(AgsMidiParser *midi_parser, guint status);

  xmlNode* (*change_parameter)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*change_pitch_bend)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*change_program)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*change_channel_pressure)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*not_defined)(AgsMidiParser *midi_parser, guint status);

  xmlNode* (*sysex)(AgsMidiParser *midi_parser, guint status);
  xmlNode* (*system_common)(AgsMidiParser *midi_parser, guint status);

  xmlNode* (*meta_event)(AgsMidiParser *midi_parser, guint status);
```

```
  xmlNode* (*sequence_number)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*end_of_track)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*smtpe)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*tempo)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*time_signature)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*key_signature)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*sequencer_meta_event)(AgsMidiParser *midi_parser, guint meta_type);
  xmlNode* (*text_event)(AgsMidiParser *midi_parser, guint meta_type);
};
```

Property Details

The "file" property

```
  "file"                      gpointer
```

The file to parse data from.

Owner: AgsMidiParser

Flags: Read / Write

Since: 3.0.0

Signal Details

The "change-channel-pressure" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::change-channel-pressure signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "change-parameter" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::change-parameter signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "change-pitch-bend" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::change-pitch-bend signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "change-program" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::change-program signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "end-of-track" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::end-of-track signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "key-off" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::key-off signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "key-on" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::key-on signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "key-pressure" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::key-pressure signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "key-signature" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::key-signature signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "meta-event" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::meta-event signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "midi-getc" signal

```
int
user_function (AgsMidiParser *midi_parser,
               gpointer       user_data)
```

The ::midi-getc signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
user_data	user data set when the signal handler was connected.	

Returns

The character read

Flags: Run Last

Since: 3.0.0

The "not-defined" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::not-defined signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "on-error" signal

```
void
user_function (AgsMidiParser *midi_parser,
               gpointer       error,
               gpointer       user_data)
```

The ::on-error signal is emited as error occurs.

Parameters

midi_parser	the AgsMidiParser	
error	the return location of GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "parse-bytes" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               gpointer       buffer,
               guint          buffer_length,
               gpointer       user_data)
```

The ::parse-bytes signal is emited during parsing of bytes.

Parameters

midi_parser	the AgsMidiParser	
buffer	the MIDI data	
buffer_length	the buffer's length	

user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "parse-full" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               gpointer       user_data)
```

The ::parse-full signal is emited during parsing of midi file.

Parameters

midi_parser	the AgsMidiParser	
user_data	user data set when the signal handler was connected.	

Returns

The XML doc

Flags: Run Last

Since: 3.0.0

The "parse-header" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               gpointer       user_data)
```

The ::parse-header signal is emited during parsing of header.

Parameters

midi_parser	the AgsMidiParser	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the header

Flags: Run Last

Since: 3.0.0

The "parse-track" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               gpointer       user_data)
```

The ::parse-track signal is emited during parsing of track.

Parameters

midi_parser	the AgsMidiParser	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the track

Flags: Run Last

Since: 3.0.0

The "sequence-number" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::sequence-number signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "sequencer-meta-event" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::sequencer-meta-event signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "smtpe" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::smtpe signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "sysex" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::sysex signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "system-common" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::system-common signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "tempo" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::tempo signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "text-event" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::text-event signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "time-signature" signal

```
gpointer
user_function (AgsMidiParser *midi_parser,
               guint          status,
               gpointer       user_data)
```

The ::time-signature signal is emited during parsing of event.

Parameters

midi_parser	the AgsMidiParser	
status	the MIDI status byte	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

Chapter 14

Audio OSC - OSC support

The Open Sound Control Content format.

14.1 AgsOscBufferUtil

AgsOscBufferUtil — OSC buffer util

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_osc_buffer_util_put_int32 ()
void	ags_osc_buffer_util_get_int32 ()
void	ags_osc_buffer_util_put_timetag ()
void	ags_osc_buffer_util_get_timetag ()
void	ags_osc_buffer_util_put_float ()
void	ags_osc_buffer_util_get_float ()
void	ags_osc_buffer_util_put_string ()
void	ags_osc_buffer_util_get_string ()
void	ags_osc_buffer_util_put_blob ()
void	ags_osc_buffer_util_get_blob ()
void	ags_osc_buffer_util_put_int64 ()
void	ags_osc_buffer_util_get_int64 ()
void	ags_osc_buffer_util_put_double ()
void	ags_osc_buffer_util_get_double ()
void	ags_osc_buffer_util_put_char ()
void	ags_osc_buffer_util_get_char ()
void	ags_osc_buffer_util_put_rgba ()
void	ags_osc_buffer_util_get_rgba ()
void	ags_osc_buffer_util_put_midi ()
void	ags_osc_buffer_util_get_midi ()
void	ags_osc_buffer_util_put_packet ()
void	ags_osc_buffer_util_get_packet ()
void	ags_osc_buffer_util_put_packets ()
void	ags_osc_buffer_util_put_message ()

void	ags_osc_buffer_util_get_message ()
void	ags_osc_buffer_util_put_bundle ()
void	ags_osc_buffer_util_get_bundle ()
GType	ags_osc_buffer_util_get_type ()

Types and Values

#define	AGS_TYPE_OSC_BUFFER_UTIL
struct	AgsOscBufferUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsOscBufferUtil
```

Includes

```
#include <ags/audio/osc/ags_osc_buffer_util.h>
```

Description

Utility functions for OSC buffer.

Functions

ags_osc_buffer_util_put_int32 ()

```
void
ags_osc_buffer_util_put_int32 (guchar *buffer,
                               gint32 val);
```

Put *val* in *buffer* .

Parameters

buffer	the guchar buffer	
val	the 32 bit integer value	

Since: 3.0.0

ags_osc_buffer_util_get_int32 ()

```
void
ags_osc_buffer_util_get_int32 (guchar *buffer,
                               gint32 *val);
```

Get *val* from *buffer* .

Parameters

buffer	the guchar buffer	
val	return location of the 32 bit integer value	

Since: 3.0.0

ags_osc_buffer_util_put_timetag ()

```
void
ags_osc_buffer_util_put_timetag (guchar *buffer,
                                 gint32 tv_secs,
                                 gint32 tv_fraction,
                                 gboolean immediately);
```

Put timetag specified by *tv_secs* , *tv_fraction* and *immediately* to *buffer* .

Parameters

buffer	the guchar buffer	
tv_secs	number of seconds since midnight on January 1, 1900	
tv_fraction	fraction of seconds to a precision of about 200 picoseconds	
immediately	if TRUE apply immediately, otherwise FALSE not immediately	

Since: 3.0.0

ags_osc_buffer_util_get_timetag ()

```
void
ags_osc_buffer_util_get_timetag (guchar *buffer,
                                 gint32 *tv_secs,
                                 gint32 *tv_fraction,
                                 gboolean *immediately);
```

Get timetag specified by *tv_secs* , *tv_fraction* and *immediately* from *buffer* .

Parameters

buffer	the guchar buffer	
tv_secs	return location of number of seconds since midnight on January 1, 1900	
tv_fraction	return location of fraction of seconds to a precision of about 200 picoseconds	

immediately	return location of if TRUE apply immediately, otherwise FALSE not immediately	

Since: 3.0.0

ags_osc_buffer_util_put_float ()

```
void
ags_osc_buffer_util_put_float (guchar *buffer,
                               gfloat val);
```

Put *val* to *buffer* .

Parameters

buffer	the guchar buffer	
val	the float value	

Since: 3.0.0

ags_osc_buffer_util_get_float ()

```
void
ags_osc_buffer_util_get_float (guchar *buffer,
                               gfloat *val);
```

Get *val* from *buffer* .

Parameters

buffer	the guchar buffer	
val	the float value	

Since: 3.0.0

ags_osc_buffer_util_put_string ()

```
void
ags_osc_buffer_util_put_string (guchar *buffer,
                                gchar *str,
                                gsize length);
```

Put *str* to *buffer* .

Parameters

buffer	the guchar buffer	
str	the ASCII string	
length	the length to write or -1 to write until null byte	

Since: 3.0.0

ags_osc_buffer_util_get_string ()

```
void
ags_osc_buffer_util_get_string (guchar *buffer,
                                gchar **str,
                                gsize *length);
```

Get *str* from *buffer* .

Parameters

buffer	the guchar buffer	
str	the return location of ASCII string	
length	the return location of bytes read	

Since: 3.0.0

ags_osc_buffer_util_put_blob ()

```
void
ags_osc_buffer_util_put_blob (guchar *buffer,
                              gint32 data_size,
                              guchar *data);
```

Put *data* to *buffer* .

Parameters

buffer	the guchar buffer	
data_size	the blob's data size	
data	the blob's data	

Since: 3.0.0

ags_osc_buffer_util_get_blob ()

```
void
ags_osc_buffer_util_get_blob (guchar *buffer,
                              gint32 *data_size,
                              guchar **data);
```

Get *data* from *buffer* .

Parameters

buffer	the guchar buffer	
data_size	the return location of blob's data size	
data	the return location of blob's data	

Since: 3.0.0

ags_osc_buffer_util_put_int64 ()

```
void
ags_osc_buffer_util_put_int64 (guchar *buffer,
                               gint64 val);
```

Put *val* to *buffer* .

Parameters

buffer	the guchar buffer	
val	the 64 bit integer	

Since: 3.0.0

ags_osc_buffer_util_get_int64 ()

```
void
ags_osc_buffer_util_get_int64 (guchar *buffer,
                               gint64 *val);
```

Get *val* from *buffer* .

Parameters

buffer	the guchar buffer	
val	the return location of 64 bit integer	

Since: 3.0.0

ags_osc_buffer_util_put_double ()

```
void
ags_osc_buffer_util_put_double (guchar *buffer,
                                gdouble val);
```

Put *val* to *buffer* .

Parameters

buffer	the guchar buffer	
val	the double floating point value	

Since: 3.0.0

ags_osc_buffer_util_get_double ()

```
void
ags_osc_buffer_util_get_double (guchar *buffer,
                                gdouble *val);
```

Get *val* from *buffer* .

Parameters

buffer	the guchar buffer	
val	the return location of double floating point value	

Since: 3.0.0

ags_osc_buffer_util_put_char ()

```
void
ags_osc_buffer_util_put_char (guchar *buffer,
                              gchar val);
```

Put *val* to *buffer* .

Parameters

buffer	the guchar buffer	
val	the ASCII char	

Since: 3.0.0

ags_osc_buffer_util_get_char ()

```
void
ags_osc_buffer_util_get_char (guchar *buffer,
                              gchar *val);
```

Get *val* from *buffer* .

Parameters

buffer	the guchar buffer	
val	the return location of ASCII char	

Since: 3.0.0

ags_osc_buffer_util_put_rgba ()

```
void
ags_osc_buffer_util_put_rgba (guchar *buffer,
                              guint8 r,
                              guint8 g,
                              guint8 b,
                              guint8 a);
```

Put RGBA color specified by *r* , *g* , *b* and *a* to *buffer* .

Parameters

buffer	the guchar buffer	
r	the red value	
g	the green value	
b	the blue value	
a	the alpha value	

Since: 3.0.0

ags_osc_buffer_util_get_rgba ()

```
void
ags_osc_buffer_util_get_rgba (guchar *buffer,
                              guint8 *r,
                              guint8 *g,
                              guint8 *b,
                              guint8 *a);
```

Get RGBA color specified by *r* , *g* , *b* and *a* from *buffer* .

Parameters

buffer	the guchar buffer	
r	the return location of red value	
g	the return location of green value	
b	the return location of blue value	
a	the return location of alpha value	

Since: 3.0.0

ags_osc_buffer_util_put_midi ()

```
void
ags_osc_buffer_util_put_midi (guchar *buffer,
                              guint8 port,
                              guint8 status_byte,
                              guint8 data0,
                              guint8 data1);
```

Put MIDI to *buffer* .

Parameters

buffer	the guchar buffer	
port	the port	
status_byte	the status byte	
data0	the first piece of data	
data1	the second piece of data	

Since: 3.0.0

ags_osc_buffer_util_get_midi ()

```
void
ags_osc_buffer_util_get_midi (guchar *buffer,
                              guint8 *port,
                              guint8 *status_byte,
                              guint8 *data0,
                              guint8 *data1);
```

Get MIDI from *buffer* .

Parameters

buffer	the guchar buffer	
port	the return location of port	
status_byte	the return location of status byte	
data0	the return location of first piece of data	
data1	the return location of second piece of data	

Since: 3.0.0

ags_osc_buffer_util_put_packet ()

```
void
ags_osc_buffer_util_put_packet (guchar *buffer,
                                gint32 packet_size,
                                guchar *packet);
```

Put *packet* to *buffer* .

Parameters

buffer	the guchar buffer	
packet_size	the packet's size	
packet	the packet	

Since: 3.0.0

ags_osc_buffer_util_get_packet ()

```
void
ags_osc_buffer_util_get_packet (guchar *buffer,
                                gint32 *packet_size,
                                guchar **packet);
```

Get *packet* from *buffer* .

Parameters

buffer	the guchar buffer	
packet_size	the return location of packet's size	
packet	the return location of packet	

Since: 3.0.0

ags_osc_buffer_util_put_packets ()

```
void
ags_osc_buffer_util_put_packets (guchar *buffer,
                                 gint32 packet_size,
                                 ...);
```

Put packets to *buffer* .

Parameters

buffer	the guchar buffer	
packet_size	the packet size or -1 if no packet	
...	the packets followed by packet size until it is -1	

Since: 3.0.0

ags_osc_buffer_util_put_message ()

```
void
ags_osc_buffer_util_put_message (guchar *buffer,
                                 gchar *address_pattern,
                                 gchar *type_tag);
```

Put message heading to *buffer* . Note you have to put following arguments yourself.

Parameters

buffer	the guchar buffer	
address_pattern	the address pattern	
type_tag	the type tag	

Since: 3.0.0

ags_osc_buffer_util_get_message ()

```
void
ags_osc_buffer_util_get_message (guchar *buffer,
                                 gchar **address_pattern,
                                 gchar **type_tag);
```

Get message heading from *buffer* . Note you have to get following arguments yourself.

Parameters

buffer	the guchar buffer	
address_pattern	return location of the address pattern	
type_tag	return location of the type tag	

Since: 3.0.0

ags_osc_buffer_util_put_bundle ()

```
void
ags_osc_buffer_util_put_bundle (guchar *buffer,
                                gint32 tv_secs,
                                gint32 tv_fraction,
                                gboolean immediately);
```

Put bundle to *buffer* .

Parameters

buffer	the guchar buffer	
tv_secs	number of seconds since midnight on January 1, 1900	
tv_fraction	fraction of seconds to a precision of about 200 picoseconds	
immediately	if TRUE apply immediately, otherwise FALSE not immediately	

Since: 3.0.0

ags_osc_buffer_util_get_bundle ()

```
void
ags_osc_buffer_util_get_bundle (guchar *buffer,
                                gint32 *tv_secs,
                                gint32 *tv_fraction,
                                gboolean *immediately);
```

Get bundle from *buffer* .

Parameters

buffer	the guchar buffer	
tv_secs	return location of number of seconds since midnight on January 1, 1900	
tv_fraction	return location of fraction of seconds to a precision of about 200 picoseconds	
immediately	return location of if TRUE apply immediately, otherwise FALSE not immediately	

Since: 3.0.0

ags_osc_buffer_util_get_type ()

```
GType
ags_osc_buffer_util_get_type (void);
```

Types and Values

AGS_TYPE_OSC_BUFFER_UTIL

```
#define AGS_TYPE_OSC_BUFFER_UTIL         (ags_osc_buffer_util_get_type())
```

struct AgsOscBufferUtil

```
struct AgsOscBufferUtil {
  //empty
};
```

14.2 AgsOscUtil

AgsOscUtil — OSC util

Stability Level

Stable, unless otherwise indicated

Functions

guint	ags_osc_util_type_tag_string_count_type ()
guchar *	ags_osc_util_meta_data ()
guchar *	ags_osc_util_mime_header ()
guchar *	ags_osc_util_slip_encode ()
guchar *	ags_osc_util_slip_decode ()
void	ags_osc_util_timetag_now ()
GType	ags_osc_util_get_type ()

Types and Values

#define	AGS_OSC_UTIL_TYPE_TAG_STRING_INT32
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_FLOAT
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_STRING
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_BLOB
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_INT64
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_TIMETAG
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_DOUBLE
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_SYMBOL
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_CHAR
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_RGBA
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_MIDI
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_TRUE
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_FALSE
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_NIL
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_INFINITE
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_START
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_END
#define	AGS_OSC_UTIL_TYPE_TAG_STRING_ALL
#define	AGS_OSC_UTIL_ADDRESS_PATTERN_VALID_CHARS_PATTERN
#define	AGS_OSC_UTIL_SLIP_CHUNK_LENGTH
#define	AGS_OSC_UTIL_SLIP_END
#define	AGS_OSC_UTIL_SLIP_ESC
#define	AGS_OSC_UTIL_SLIP_ESC_END
#define	AGS_OSC_UTIL_SLIP_ESC_ESC
#define	AGS_TYPE_OSC_UTIL
struct	AgsOscUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsOscUtil
```

Includes

```
#include <ags/audio/osc/ags_osc_util.h>
```

Description

Utility functions for OSC.

Functions

ags_osc_util_type_tag_string_count_type ()

```
guint
ags_osc_util_type_tag_string_count_type
                                (gchar *type_tag);
```

Count types in *type_tag* .

Parameters

type_tag	the type tag string

Returns

the count of types specified

Since: 3.0.0

ags_osc_util_meta_data ()

```
guchar~*
ags_osc_util_meta_data (gchar *uri,
                        guint *meta_data_length);
```

Get meta data.

Parameters

uri	the URI
meta_data_length	the return location of meta data's length

Returns

the byte array containing meta data

Since: 3.0.0

ags_osc_util_mime_header ()

```
guchar~*
ags_osc_util_mime_header (gchar *uri,
                          guint *mime_header_length);
```

Get MIME header.

Parameters

uri	the URI	
mime_header_length	the return location of MIME header's length	

Returns

the byte array containing MIME header

Since: 3.0.0

ags_osc_util_slip_encode ()

```
guchar~*
ags_osc_util_slip_encode (guchar *osc_buffer,
                          guint buffer_length,
                          guint *returned_buffer_length);
```

Encode *osc_buffer* to be suitable for network transmission. See SLIP (RFC1055).

Parameters

osc_buffer	the OSC buffer	
buffer_length	the OSC buffer's length	
returned_buffer_length	the returned buffer's length	

Returns

the encoded byte array

Since: 3.0.0

ags_osc_util_slip_decode ()

```
guchar~*
ags_osc_util_slip_decode (guchar *slip_buffer,
                          guint slip_buffer_length,
                          guint *returned_buffer_length);
```

Decode *slip_buffer* from SLIP encoded format. See SLIP (RFC1055).

Parameters

slip_buffer	the SLIP encoded OSC buffer	
slip_buffer_length	the buffer length of SLIP encoded OSC buffer	
returned_buffer_length	the returned buffer's length	

Returns

the OSC buffer as byte array

Since: 3.0.0

ags_osc_util_timetag_now ()

```
void
ags_osc_util_timetag_now (gint32 *tv_sec,
                          gint32 *tv_fraction);
```

Get current time.

Parameters

tv_sec	the return location of number of seconds since midnight on January 1, 1900	
tv_fraction	the return location of fraction of seconds to a precision of about 200 picoseconds	

Since: 3.0.0

ags_osc_util_get_type ()

```
GType
ags_osc_util_get_type (void);
```

Types and Values

AGS_OSC_UTIL_TYPE_TAG_STRING_INT32

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_INT32 'i'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_FLOAT

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_FLOAT 'f'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_STRING

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_STRING 's'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_BLOB

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_BLOB 'b'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_INT64

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_INT64 'h'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_TIMETAG

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_TIMETAG 't'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_DOUBLE

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_DOUBLE 'd'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_SYMBOL

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_SYMBOL 'S'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_CHAR

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_CHAR 'c'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_RGBA

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_RGBA 'r'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_MIDI

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_MIDI 'm'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_TRUE

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_TRUE 'T'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_FALSE

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_FALSE 'F'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_NIL

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_NIL 'N'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_INFINITE

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_INFINITE 'I'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_START

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_START '['
```

AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_END

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_END ']'
```

AGS_OSC_UTIL_TYPE_TAG_STRING_ALL

```
#define AGS_OSC_UTIL_TYPE_TAG_STRING_ALL "ifshtdScrmTFNI"
```

AGS_OSC_UTIL_ADDRESS_PATTERN_VALID_CHARS_PATTERN

```
#define AGS_OSC_UTIL_ADDRESS_PATTERN_VALID_CHARS_PATTERN "([a-zA-Z  ↩
    \\#\\*\\,\\/\\?\\[\\]\\{\\}]+)"
```

AGS_OSC_UTIL_SLIP_CHUNK_LENGTH

```
#define AGS_OSC_UTIL_SLIP_CHUNK_LENGTH (8192)
```

AGS_OSC_UTIL_SLIP_END

```
#define AGS_OSC_UTIL_SLIP_END (192)
```

AGS_OSC_UTIL_SLIP_ESC

```
#define AGS_OSC_UTIL_SLIP_ESC (219)
```

AGS_OSC_UTIL_SLIP_ESC_END

```
#define AGS_OSC_UTIL_SLIP_ESC_END (220)
```

AGS_OSC_UTIL_SLIP_ESC_ESC

```
#define AGS_OSC_UTIL_SLIP_ESC_ESC (221)
```

AGS_TYPE_OSC_UTIL

```
#define AGS_TYPE_OSC_UTIL         (ags_osc_util_get_type())
```

struct AgsOscUtil

```
struct AgsOscUtil {
  //empty
};
```

14.3 AgsOscBuilder

AgsOscBuilder — OSC builder

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_BUILDER_GET_OBJ_MUTEX()
#define	AGS_OSC_BUILDER_PACKET()
#define	AGS_OSC_BUILDER_BUNDLE()
#define	AGS_OSC_BUILDER_MESSAGE()
AgsOscBuilderPacket *	ags_osc_builder_packet_alloc ()
void	ags_osc_builder_packet_free ()
AgsOscBuilderBundle *	ags_osc_builder_bundle_alloc ()
void	ags_osc_builder_bundle_free ()
AgsOscBuilderMessage *	ags_osc_builder_message_alloc ()
void	ags_osc_builder_message_free ()
void	ags_osc_builder_osc_putc ()
void	ags_osc_builder_on_error ()
void	ags_osc_builder_append_packet ()
void	ags_osc_builder_append_bundle ()
void	ags_osc_builder_append_message ()
void	ags_osc_builder_append_value ()
void	ags_osc_builder_build ()
guchar *	ags_osc_builder_get_data ()
guchar *	ags_osc_builder_get_data_with_length ()
AgsOscBuilder *	ags_osc_builder_new ()
#define	AGS_IS_OSC_BUILDER()
#define	AGS_IS_OSC_BUILDER_CLASS()
#define	AGS_OSC_BUILDER()
#define	AGS_OSC_BUILDER_CLASS()
#define	AGS_OSC_BUILDER_GET_CLASS()
GType	ags_osc_builder_get_type ()

Signals

void	append-bundle	Run Last
void	append-message	Run Last
void	append-packet	Run Last
void	append-value	Run Last
void	on-error	Run Last
void	osc-putc	Run Last

Types and Values

#define	AGS_OSC_BUILDER_DEFAULT_CHUNK_SIZE
#define	AGS_OSC_BUILDER_MESSAGE_DEFAULT_CHUNK_SIZE
enum	AgsOscBuilderFlags

struct	AgsOscBuilderPacket
struct	AgsOscBuilderBundle
struct	AgsOscBuilderMessage
#define	AGS_TYPE_OSC_BUILDER
struct	AgsOscBuilder
struct	AgsOscBuilderClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscBuilder
```

Includes

```
#include <ags/audio/osc/ags_osc_builder.h>
```

Description

The AgsOscBuilder lets you create OSC files using its builder functions.

Functions

AGS_OSC_BUILDER_GET_OBJ_MUTEX()

```
#define AGS_OSC_BUILDER_GET_OBJ_MUTEX(obj) (&(((AgsOscBuilder *) obj)->obj_mutex))
```

AGS_OSC_BUILDER_PACKET()

```
#define AGS_OSC_BUILDER_PACKET(x) ((AgsOscBuilderPacket *)(x))
```

AGS_OSC_BUILDER_BUNDLE()

```
#define AGS_OSC_BUILDER_BUNDLE(x) ((AgsOscBuilderBundle *)(x))
```

AGS_OSC_BUILDER_MESSAGE()

```
#define AGS_OSC_BUILDER_MESSAGE(x) ((AgsOscBuilderMessage *)(x))
```

ags_osc_builder_packet_alloc ()

```
AgsOscBuilderPacket~*
ags_osc_builder_packet_alloc (guint64 offset);
```

Allocate AgsOscBuilderPacket.

Parameters

offset	the current offset

Returns

the newly allocated AgsOscBuilderPacket.

[type gpointer][transfer none]

Since: 3.0.0

ags_osc_builder_packet_free ()

```
void
ags_osc_builder_packet_free (AgsOscBuilderPacket *packet);
```

Free `packet` .

Parameters

packet	the AgsOscBuilderPacket.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_osc_builder_bundle_alloc ()

```
AgsOscBuilderBundle~*
ags_osc_builder_bundle_alloc (guint64 offset);
```

Allocate AgsOscBuilderBundle.

Parameters

offset	the current offset

Returns

the newly allocated AgsOscBuilderBundle.

[type gpointer][transfer none]

Since: 3.0.0

ags_osc_builder_bundle_free ()

```
void
ags_osc_builder_bundle_free (AgsOscBuilderBundle *bundle);
```

Free `bundle` .

Parameters

bundle	the AgsOscBuilderBundle.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_osc_builder_message_alloc ()

```
AgsOscBuilderMessage~*
ags_osc_builder_message_alloc (guint64 offset);
```

Allocate AgsOscBuilderMessage.

Parameters

offset	the current offset	

Returns

the newly allocated AgsOscBuilderMessage.

[type gpointer][transfer none]

Since: 3.0.0

ags_osc_builder_message_free ()

```
void
ags_osc_builder_message_free (AgsOscBuilderMessage *message);
```

Free *message* .

Parameters

message	the AgsOscBuilderMessage.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_osc_builder_osc_putc ()

```
void
ags_osc_builder_osc_putc (AgsOscBuilder *osc_builder,
                          gint c);
```

Put character

Parameters

osc_builder	the AgsOscBuilder	
c	the character	

Since: 3.0.0

ags_osc_builder_on_error ()

```
void
ags_osc_builder_on_error (AgsOscBuilder *osc_builder,
                          GError **error);
```

Report error.

Parameters

osc_builder	the AgsOscBuilder	
error	the return location of GError	

Since: 3.0.0

ags_osc_builder_append_packet ()

```
void
ags_osc_builder_append_packet (AgsOscBuilder *osc_builder);
```

Parameters

osc_builder	the AgsOscBuilder	

Since: 3.0.0

ags_osc_builder_append_bundle ()

```
void
ags_osc_builder_append_bundle (AgsOscBuilder *osc_builder,
                               AgsOscBuilderBundle *parent_bundle,
                               gint tv_secs,
                               gint tv_fraction,
                               gboolean immediately);
```

Append bundle.

Parameters

osc_builder	the AgsOscBuilder	
parent_bundle	the parent AgsOscBuilderBundle or NULL	
tv_secs	number of seconds since midnight on January 1, 1900	
tv_fraction	fraction of seconds to a precision of about 200 picoseconds	
immediately	if TRUE apply immediately, otherwise FALSE not immediately	

Since: 3.0.0

ags_osc_builder_append_message ()

```
void
ags_osc_builder_append_message (AgsOscBuilder *osc_builder,
                                AgsOscBuilderBundle *parent_bundle,
                                gchar *address_pattern,
                                gchar *type_tag);
```

Append message.

Parameters

osc_builder	the AgsOscBuilder	
parent_bundle	the parent AgsOscBuilderBundle or NULL	
address_pattern	the address pattern	
type_tag	the type tag	

Since: 3.0.0

ags_osc_builder_append_value ()

```
void
ags_osc_builder_append_value (AgsOscBuilder *osc_builder,
                              AgsOscBuilderMessage *message,
                              gint v_type,
                              GValue *value);
```

Append value.

Parameters

osc_builder	the AgsOscBuilder	
message	the message	
v_type	the type as char	
value	the GValue containinig value	

Since: 3.0.0

ags_osc_builder_build ()

```
void
ags_osc_builder_build (AgsOscBuilder *osc_builder);
```

Build the OSC data.

Parameters

osc_builder	the AgsOscBuilder	

Since: 3.0.0

ags_osc_builder_get_data ()

```
guchar~*
ags_osc_builder_get_data (AgsOscBuilder *osc_builder);
```

Get OSC data of *osc_builder* .

Parameters

osc_builder	the AgsOscBuilder	

Returns

the OSC data

Since: 3.7.24

ags_osc_builder_get_data_with_length ()

```
guchar~*
ags_osc_builder_get_data_with_length (AgsOscBuilder *osc_builder,
                                      guint *length);
```

Get OSC data of *osc_builder* .

Parameters

osc_builder	the AgsOscBuilder	
length	the length of data.	*[out]*

Returns

the OSC data as array.

[transfer full]

Since: 3.7.24

ags_osc_builder_new ()

```
AgsOscBuilder~*
ags_osc_builder_new ();
```

Creates a new instance of AgsOscBuilder

Returns

the new AgsOscBuilder

Since: 3.0.0

AGS_IS_OSC_BUILDER()

```
#define AGS_IS_OSC_BUILDER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_BUILDER))
```

AGS_IS_OSC_BUILDER_CLASS()

```
#define AGS_IS_OSC_BUILDER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_BUILDER))
```

AGS_OSC_BUILDER()

```
#define AGS_OSC_BUILDER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_BUILDER, AgsOscBuilder))
```

AGS_OSC_BUILDER_CLASS()

```
#define AGS_OSC_BUILDER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_BUILDER, AgsOscBuilderClass))
```

AGS_OSC_BUILDER_GET_CLASS()

```
#define AGS_OSC_BUILDER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_BUILDER, AgsOscBuilderClass))
```

ags_osc_builder_get_type ()

```
GType
ags_osc_builder_get_type (void);
```

Types and Values

AGS_OSC_BUILDER_DEFAULT_CHUNK_SIZE

```
#define AGS_OSC_BUILDER_DEFAULT_CHUNK_SIZE (8192)
```

AGS_OSC_BUILDER_MESSAGE_DEFAULT_CHUNK_SIZE

```
#define AGS_OSC_BUILDER_MESSAGE_DEFAULT_CHUNK_SIZE (8192)
```

enum AgsOscBuilderFlags

Members

AGS_OSC_BUILDER_ADD_VALUE	

struct AgsOscBuilderPacket

```
struct AgsOscBuilderPacket {
  guint64 offset;

  gsize packet_size;

  AgsOscBuilder *builder;

  GList *message;
  GList *bundle;
};
```

struct AgsOscBuilderBundle

```
struct AgsOscBuilderBundle {
  guint64 offset;

  gsize bundle_size;

  gint32 tv_secs;
  gint32 tv_fraction;
  gboolean immediately;

  AgsOscBuilderPacket *packet;
  AgsOscBuilderBundle *parent_bundle;

  GList *message;
  GList *bundle;
};
```

struct AgsOscBuilderMessage

```
struct AgsOscBuilderMessage {
  guint64 offset;

  gchar *address_pattern;
  gchar *type_tag;

  gsize data_allocated_length;
  gsize data_length;
  guchar *data;

  AgsOscBuilderPacket *packet;
  AgsOscBuilderBundle *parent_bundle;
};
```

AGS_TYPE_OSC_BUILDER

```
#define AGS_TYPE_OSC_BUILDER                (ags_osc_builder_get_type ())
```

struct AgsOscBuilder

```
struct AgsOscBuilder;
```

struct AgsOscBuilderClass

```
struct AgsOscBuilderClass {
  GObjectClass gobject;

  void (*osc_putc)(AgsOscBuilder *osc_builder,
       gint c);
  void (*on_error)(AgsOscBuilder *osc_builder,
       GError **error);

  /* */
  void (*append_packet)(AgsOscBuilder *osc_builder);

  void (*append_bundle)(AgsOscBuilder *osc_builder,
      AgsOscBuilderBundle *parent_bundle,
      gint tv_secs, gint tv_fraction, gboolean immediately);

  void (*append_message)(AgsOscBuilder *osc_builder,
       AgsOscBuilderBundle *parent_bundle,
       gchar *address_pattern,
       gchar *type_tag);

  void (*append_value)(AgsOscBuilder *osc_builder,
           AgsOscBuilderMessage *message,
           gint v_type,
           GValue *value);
};
```

Signal Details

The "append-bundle" signal

```
void
user_function (AgsOscBuilder *osc_builder,
               gpointer       parent_bundle,
               int            tv_secs,
               int            tv_fraction,
               gboolean       immediately,
               gpointer       user_data)
```

The ::append-bundle signal is emited during building bundle.

Parameters

osc_builder	the builder	
parent_bundle	the parent AgsOscBuilderBundle or NULL	
tv_secs	time value secondes since midnight January 1900	
tv_fraction	time value fractions of second	
immediately	if TRUE apply immediately, otherwise FALSE	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "append-message" signal

```
void
user_function (AgsOscBuilder *osc_builder,
               gpointer       parent_bundle,
               gpointer       address_pattern,
               gpointer       type_tag,
               gpointer       user_data)
```

The ::append-message signal is emited during building message.

Parameters

osc_builder	the builder	
parent_bundle	the parent AgsOscBuilderBundle or NULL	
address_pattern	the address pattern string	
type_tag	the type tag string	

user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "append-packet" signal

```
void
user_function (AgsOscBuilder *osc_builder,
               gpointer       user_data)
```

The ::append-packet signal is emited during building packet.

Parameters

osc_builder	the builder	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "append-value" signal

```
void
user_function (AgsOscBuilder *osc_builder,
               int            message,
               gpointer       v_type,
               gpointer       user_data)
```

The ::append-value signal is emited during building value.

Parameters

osc_builder	the builder	
message	the message	
v_type	the type as ASCII char	
value	the GValue	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "on-error" signal

```
void
user_function (AgsOscBuilder *osc_builder,
               gpointer       error,
               gpointer       user_data)
```

The ::on-error signal is emited during building of event.

Parameters

osc_builder	the builder	
error	the GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "osc-putc" signal

```
void
user_function (AgsOscBuilder *osc_builder,
               int            error,
               gpointer       user_data)
```

The ::osc-putc signal is emited during putting char to file.

Parameters

osc_builder	the builder	
error	the GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

14.4 AgsOscParser

AgsOscParser — the OSC parser

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_PARSER_GET_OBJ_MUTEX()
gint32	ags_osc_parser_read_gint32 ()
gint64	ags_osc_parser_read_gint64 ()
gfloat	ags_osc_parser_read_gfloat ()
gdouble	ags_osc_parser_read_gdouble ()
gchar *	ags_osc_parser_read_text ()
int	ags_osc_parser_osc_getc ()
void	ags_osc_parser_on_error ()
xmlDoc *	ags_osc_parser_parse_full ()
xmlNode *	ags_osc_parser_parse_bytes ()
xmlNode *	ags_osc_parser_packet ()
xmlNode *	ags_osc_parser_bundle ()
xmlNode *	ags_osc_parser_message ()
xmlNode *	ags_osc_parser_value ()
AgsOscParser *	ags_osc_parser_new ()
#define	AGS_IS_OSC_PARSER()
#define	AGS_IS_OSC_PARSER_CLASS()
#define	AGS_OSC_PARSER()
#define	AGS_OSC_PARSER_CLASS()
#define	AGS_OSC_PARSER_GET_CLASS()
GType	ags_osc_parser_get_type ()

Signals

gpointer	bundle	Run Last
gpointer	message	Run Last
void	on-error	Run Last
int	osc-getc	Run Last
gpointer	packet	Run Last
gpointer	parse-bytes	Run Last
gpointer	parse-full	Run Last
gpointer	value	Run Last

Types and Values

#define	AGS_OSC_PARSER_MAX_TEXT_LENGTH
enum	AgsOscParserFlags
#define	AGS_TYPE_OSC_PARSER
struct	AgsOscParser
struct	AgsOscParserClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsOscParser
```

Includes

```
#include <ags/audio/osc/ags_osc_parser.h>
```

Description

AgsOscParser reads your osc parsers.

Functions

AGS_OSC_PARSER_GET_OBJ_MUTEX()

```
#define AGS_OSC_PARSER_GET_OBJ_MUTEX(obj) (&(((AgsOscParser *) obj)->obj_mutex))
```

ags_osc_parser_read_gint32 ()

```
gint32
ags_osc_parser_read_gint32 (AgsOscParser *osc_parser);
```

Read 32 bit integer.

Parameters

osc_parser	the AgsOscParser

Returns

the gint32 read

Since: 3.0.0

ags_osc_parser_read_gint64 ()

```
gint64
ags_osc_parser_read_gint64 (AgsOscParser *osc_parser);
```

Read 64 bit integer.

Parameters

osc_parser	the AgsOscParser

Returns

the gint64 read

Since: 3.0.0

ags_osc_parser_read_gfloat ()

```
gfloat
ags_osc_parser_read_gfloat (AgsOscParser *osc_parser);
```

Read floating point value.

Parameters

osc_parser	the AgsOscParser

Returns

the gfloat read

Since: 3.0.0

ags_osc_parser_read_gdouble ()

```
gdouble
ags_osc_parser_read_gdouble (AgsOscParser *osc_parser);
```

Read double precision floating point value.

Parameters

osc_parser	the AgsOscParser

Returns

the gdouble read

Since: 3.0.0

ags_osc_parser_read_text ()

```
gchar~*
ags_osc_parser_read_text (AgsOscParser *osc_parser,
                          gint length);
```

Read text.

Parameters

osc_parser	the AgsOscParser
length	the length

Returns

the text read as string

Since: 3.0.0

ags_osc_parser_osc_getc ()

```
int
ags_osc_parser_osc_getc (AgsOscParser *osc_parser);
```

Read byte.

Parameters

osc_parser	the AgsOscParser

Returns

the byte read

Since: 3.0.0

ags_osc_parser_on_error ()

```
void
ags_osc_parser_on_error (AgsOscParser *osc_parser,
                         GError **error);
```

On error event.

Parameters

osc_parser	the AgsOscParser
error	the GError return location

Since: 3.0.0

ags_osc_parser_parse_full ()

```
xmlDoc~*
ags_osc_parser_parse_full (AgsOscParser *osc_parser);
```

Parse full document.

Parameters

osc_parser	the AgsOscParser

Returns

the parsed XML doc.

[transfer none]

Since: 3.0.0

ags_osc_parser_parse_bytes ()

```
xmlNode~*
ags_osc_parser_parse_bytes (AgsOscParser *osc_parser,
                            guchar *osc_buffer,
                            guint buffer_length);
```

Parse bytes.

Parameters

osc_parser	the AgsOscParser	
osc_buffer	the data buffer	
buffer_length	the length of data buffer	

Returns

the parsed XML node.

[transfer none]

Since: 3.0.0

ags_osc_parser_packet ()

```
xmlNode~*
ags_osc_parser_packet (AgsOscParser *osc_parser);
```

Parse OSC packet.

Parameters

osc_parser	the AgsOscParser	

Returns

the parsed XML node.

[transfer none]

Since: 3.0.0

ags_osc_parser_bundle ()

```
xmlNode~*
ags_osc_parser_bundle (AgsOscParser *osc_parser);
```

Parse OSC bundle.

Parameters

osc_parser	the AgsOscParser	

Returns

the parsed XML node.

[transfer none]

Since: 3.0.0

ags_osc_parser_message ()

```
xmlNode~*
ags_osc_parser_message (AgsOscParser *osc_parser);
```

Parse OSC message.

Parameters

osc_parser	the AgsOscParser

Returns

the parsed XML node.

[transfer none]

Since: 3.0.0

ags_osc_parser_value ()

```
xmlNode~*
ags_osc_parser_value (AgsOscParser *osc_parser,
                      guint v_type);
```

Parse OSC value.

Parameters

osc_parser	the AgsOscParser	
v_type	value type	

Returns

the parsed XML node.

[transfer none]

Since: 3.0.0

ags_osc_parser_new ()

```
AgsOscParser~*
ags_osc_parser_new ();
```

Creates a new instance of AgsOscParser

Returns

the new AgsOscParser

Since: 3.0.0

AGS_IS_OSC_PARSER()

```
#define AGS_IS_OSC_PARSER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_PARSER))
```

AGS_IS_OSC_PARSER_CLASS()

```
#define AGS_IS_OSC_PARSER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_PARSER))
```

AGS_OSC_PARSER()

```
#define AGS_OSC_PARSER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_PARSER, AgsOscParser))
```

AGS_OSC_PARSER_CLASS()

```
#define AGS_OSC_PARSER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_PARSER, AgsOscParserClass))
```

AGS_OSC_PARSER_GET_CLASS()

```
#define AGS_OSC_PARSER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_PARSER, AgsOscParserClass))
```

ags_osc_parser_get_type ()

```
GType
ags_osc_parser_get_type (void);
```

Types and Values

AGS_OSC_PARSER_MAX_TEXT_LENGTH

```
#define AGS_OSC_PARSER_MAX_TEXT_LENGTH (4096)
```

enum AgsOscParserFlags

Enum values to control the behavior or indicate internal state of AgsOscParser by enable/disable as flags.

Members

AGS_OSC_PARSER_EOF	the parser reached EOF

AGS_TYPE_OSC_PARSER

```
#define AGS_TYPE_OSC_PARSER                (ags_osc_parser_get_type ())
```

struct AgsOscParser

```
struct AgsOscParser;
```

struct AgsOscParserClass

```
struct AgsOscParserClass {
  GObjectClass gobject;

  int (*osc_getc)(AgsOscParser *osc_parser);
  void (*on_error)(AgsOscParser *osc_parser,
       GError **error);

  xmlDoc* (*parse_full)(AgsOscParser *osc_parser);
  xmlNode* (*parse_bytes)(AgsOscParser *osc_parser,
        guchar *osc_buffer,
        guint buffer_length);

  xmlNode* (*packet)(AgsOscParser *osc_parser);

  xmlNode* (*bundle)(AgsOscParser *osc_parser);

  xmlNode* (*message)(AgsOscParser *osc_parser);

  xmlNode* (*value)(AgsOscParser *osc_parser,
        guint v_type);
};
```

Signal Details

The "bundle" signal

```
gpointer
user_function (AgsOscParser *osc_parser,
               gpointer      user_data)
```

The ::bundle signal is emited during parsing.

Parameters

osc_parser	the AgsOscParser	

user_data	user data set when the signal handler was connected.	

Returns

The XML node representing bundle

Flags: Run Last

Since: 3.0.0

The "message" signal

```
gpointer
user_function (AgsOscParser *osc_parser,
               gpointer      user_data)
```

The ::message signal is emited during parsing.

Parameters

osc_parser	the AgsOscParser	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing message

Flags: Run Last

Since: 3.0.0

The "on-error" signal

```
void
user_function (AgsOscParser *osc_parser,
               gpointer      error,
               gpointer      user_data)
```

The ::on-error signal is emited as error occurs.

Parameters

osc_parser	the AgsOscParser	
error	the return location of GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "osc-getc" signal

```
int
user_function (AgsOscParser *osc_parser,
               gpointer      user_data)
```

The ::osc-getc signal is emited during parsing of event.

Parameters

osc_parser	the AgsOscParser	
user_data	user data set when the signal handler was connected.	

Returns

The character read

Flags: Run Last

Since: 3.0.0

The "packet" signal

```
gpointer
user_function (AgsOscParser *osc_parser,
               gpointer      user_data)
```

The ::packet signal is emited during parsing.

Parameters

osc_parser	the AgsOscParser	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing packet

Flags: Run Last

Since: 3.0.0

The "parse-bytes" signal

```
gpointer
user_function (AgsOscParser *osc_parser,
               gpointer      buffer,
               guint         buffer_length,
               gpointer      user_data)
```

The ::parse-bytes signal is emited during parsing of bytes.

Parameters

osc_parser	the AgsOscParser	
buffer	the OSC data	
buffer_length	the buffer's length	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing the event

Flags: Run Last

Since: 3.0.0

The "parse-full" signal

```
gpointer
user_function (AgsOscParser *osc_parser,
               gpointer      user_data)
```

The ::parse-full signal is emited during parsing of osc file.

Parameters

osc_parser	the AgsOscParser	
user_data	user data set when the signal handler was connected.	

Returns

The XML doc

Flags: Run Last

Since: 3.0.0

The "value" signal

```
gpointer
user_function (AgsOscParser *osc_parser,
               guint         v_type,
               gpointer      user_data)
```

The ::value signal is emited during parsing.

Parameters

osc_parser	the AgsOscParser	
v_type	value type	
user_data	user data set when the signal handler was connected.	

Returns

The XML node representing value

Flags: Run Last

Since: 3.0.0

Chapter 15

Audio OSC - OSC client

The Open Sound Control client.

15.1 AgsOscClient

AgsOscClient — the OSC client

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_CLIENT_GET_OBJ_MUTEX()
gboolean	ags_osc_client_test_flags ()
void	ags_osc_client_set_flags ()
void	ags_osc_client_unset_flags ()
void	ags_osc_client_resolve ()
void	ags_osc_client_connect ()
guchar *	ags_osc_client_read_bytes ()
gboolean	ags_osc_client_write_bytes ()
AgsOscClient *	ags_osc_client_new ()
#define	AGS_IS_OSC_CLIENT()
#define	AGS_IS_OSC_CLIENT_CLASS()
#define	AGS_OSC_CLIENT()
#define	AGS_OSC_CLIENT_CLASS()
#define	AGS_OSC_CLIENT_GET_CLASS()
GType	ags_osc_client_flags_get_type ()
GType	ags_osc_client_get_type ()

Properties

char *	domain	Read / Write
char *	ip4	Read / Write
char *	ip6	Read / Write
guint	server-port	Read / Write

Signals

void	connect	Run Last
gpointer	read-bytes	Run Last
void	resolve	Run Last
gboolean	write-bytes	Run Last

Types and Values

#define	AGS_OSC_CLIENT_DEFAULT_MAX_ADDRESS_LENGTH
#define	AGS_OSC_CLIENT_DEFAULT_SERVER_PORT
#define	AGS_OSC_CLIENT_DEFAULT_DOMAIN
#define	AGS_OSC_CLIENT_DEFAULT_INET4_ADDRESS
#define	AGS_OSC_CLIENT_DEFAULT_INET6_ADDRESS
#define	AGS_OSC_CLIENT_DEFAULT_MAX_RETRY
#define	AGS_OSC_CLIENT_CHUNK_SIZE
#define	AGS_OSC_CLIENT_DEFAULT_CACHE_DATA_LENGTH
enum	AgsOscClientFlags
#define	AGS_TYPE_OSC_CLIENT
struct	AgsOscClient
struct	AgsOscClientClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOscClientFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscClient
```

Includes

```
#include <ags/audio/osc/ags_osc_client.h>
```

Description

AgsOscClient your OSC client.

Functions

AGS_OSC_CLIENT_GET_OBJ_MUTEX()

```
#define AGS_OSC_CLIENT_GET_OBJ_MUTEX(obj) (&(((AgsOscClient *) obj)->obj_mutex))
```

ags_osc_client_test_flags ()

```
gboolean
ags_osc_client_test_flags (AgsOscClient *osc_client,
                           guint flags);
```

Test *flags* to be set on *osc_client* .

Parameters

osc_client	the AgsOscClient	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_client_set_flags ()

```
void
ags_osc_client_set_flags (AgsOscClient *osc_client,
                          guint flags);
```

Set flags.

Parameters

osc_client	the AgsOscClient	
flags	the flags	

Since: 3.0.0

ags_osc_client_unset_flags ()

```
void
ags_osc_client_unset_flags (AgsOscClient *osc_client,
                            guint flags);
```

Unset flags.

Parameters

osc_client	the AgsOscClient	
flags	the flags	

Since: 3.0.0

ags_osc_client_resolve ()

```
void
ags_osc_client_resolve (AgsOscClient *osc_client);
```

Resolve OSC client.

Parameters

osc_client	the AgsOscClient

Since: 3.0.0

ags_osc_client_connect ()

```
void
ags_osc_client_connect (AgsOscClient *osc_client);
```

Connect OSC client.

Parameters

osc_client	the AgsOscClient

Since: 3.0.0

ags_osc_client_read_bytes ()

```
guchar~*
ags_osc_client_read_bytes (AgsOscClient *osc_client,
                           guint *data_length);
```

Read bytes.

Parameters

osc_client	the AgsOscClient
data_length	the return location of byte array's length

Returns

byte array read or NULL if no data available

Since: 3.0.0

ags_osc_client_write_bytes ()

```
gboolean
ags_osc_client_write_bytes (AgsOscClient *osc_client,
                            guchar *data,
                            guint data_length);
```

Write *data* using OSC client.

Parameters

osc_client	the AgsOscClient	
data	the byte array	
data_length	the length of byte array	

Returns

TRUE if all bytes written, otherwise FALSE

Since: 3.0.0

ags_osc_client_new ()

```
AgsOscClient~*
ags_osc_client_new ();
```

Creates a new instance of AgsOscClient

Returns

the new AgsOscClient

Since: 3.0.0

AGS_IS_OSC_CLIENT()

```
#define AGS_IS_OSC_CLIENT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_CLIENT))
```

AGS_IS_OSC_CLIENT_CLASS()

```
#define AGS_IS_OSC_CLIENT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_CLIENT))
```

AGS_OSC_CLIENT()

```
#define AGS_OSC_CLIENT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_CLIENT, AgsOscClient))
```

AGS_OSC_CLIENT_CLASS()

```
#define AGS_OSC_CLIENT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_CLIENT, AgsOscClientClass))
```

AGS_OSC_CLIENT_GET_CLASS()

```
#define AGS_OSC_CLIENT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_CLIENT, AgsOscClientClass))
```

ags_osc_client_flags_get_type ()

```
GType
ags_osc_client_flags_get_type ();
```

ags_osc_client_get_type ()

```
GType
ags_osc_client_get_type (void);
```

Types and Values

AGS_OSC_CLIENT_DEFAULT_MAX_ADDRESS_LENGTH

```
#define AGS_OSC_CLIENT_DEFAULT_MAX_ADDRESS_LENGTH (2048)
```

AGS_OSC_CLIENT_DEFAULT_SERVER_PORT

```
#define AGS_OSC_CLIENT_DEFAULT_SERVER_PORT (9000)
```

AGS_OSC_CLIENT_DEFAULT_DOMAIN

```
#define AGS_OSC_CLIENT_DEFAULT_DOMAIN "localhost"
```

AGS_OSC_CLIENT_DEFAULT_INET4_ADDRESS

```
#define AGS_OSC_CLIENT_DEFAULT_INET4_ADDRESS "127.0.0.1"
```

AGS_OSC_CLIENT_DEFAULT_INET6_ADDRESS

```
#define AGS_OSC_CLIENT_DEFAULT_INET6_ADDRESS "::1"
```

AGS_OSC_CLIENT_DEFAULT_MAX_RETRY

```
#define AGS_OSC_CLIENT_DEFAULT_MAX_RETRY (16)
```

AGS_OSC_CLIENT_CHUNK_SIZE

```
#define AGS_OSC_CLIENT_CHUNK_SIZE (131072)
```

AGS_OSC_CLIENT_DEFAULT_CACHE_DATA_LENGTH

```
#define AGS_OSC_CLIENT_DEFAULT_CACHE_DATA_LENGTH (256)
```

enum AgsOscClientFlags

Enum values to configure OSC client.

Members

AGS_OSC_CLIENT_INET4	use IPv4 socket
AGS_OSC_CLIENT_INET6	use IPv6 socket
AGS_OSC_CLIENT_UDP	use UDP trans-port pro-to-col
AGS_OSC_CLIENT_TCP	used TCP trans-port pro-to-col

AGS_TYPE_OSC_CLIENT

```
#define AGS_TYPE_OSC_CLIENT                (ags_osc_client_get_type ())
```

struct AgsOscClient

```
struct AgsOscClient;
```

struct AgsOscClientClass

```
struct AgsOscClientClass {
  GObjectClass gobject;

  void (*resolve)(AgsOscClient *osc_client);
  void (*connect)(AgsOscClient *osc_client);

  guchar* (*read_bytes)(AgsOscClient *osc_client,
      guint *data_length);
  gboolean (*write_bytes)(AgsOscClient *osc_client,
        guchar *data, guint data_length);
};
```

Property Details

The "domain" property

```
"domain"                    char~*
```

The domain to lookup server.

Owner: AgsOscClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ip4" property

```
"ip4"                       char~*
```

The IPv4 address as string of the server.

Owner: AgsOscClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ip6" property

```
"ip6"                       char~*
```

The IPv6 address as string of the server.

Owner: AgsOscClient

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "server-port" property

```
"server-port"               guint
```

The server port to connect.

Owner: AgsOscClient

Flags: Read / Write

Default value: 9000

Since: 3.0.0

Signal Details

The "connect" signal

```
void
user_function (AgsOscClient *osc_client,
               gpointer      user_data)
```

The ::connect signal is emited as connection is establishing.

Parameters

osc_client	the AgsOscClient	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "read-bytes" signal

```
gpointer
user_function (AgsOscClient *osc_client,
               gpointer      data_length,
               gpointer      user_data)
```

The ::read-bytes signal is emited while read bytes.

Parameters

osc_client	the AgsOscClient	
data_length	the return location of byte array's length	
user_data	user data set when the signal handler was connected.	

Returns

byte array read or NULL if no data available

Flags: Run Last

Since: 3.0.0

The "resolve" signal

```
void
user_function (AgsOscClient *osc_client,
               gpointer      user_data)
```

The ::resolve signal is emited during resolve of domain.

Parameters

osc_client	the AgsOscClient	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "write-bytes" signal

```
gboolean
user_function (AgsOscClient *osc_client,
               gpointer      data,
               guint         data_length,
               gpointer      user_data)
```

The ::write-bytes signal is emited while write bytes.

Parameters

osc_client	the AgsOscClient	
data	the byte array	
data_length	the length of byte array	
user_data	user data set when the signal handler was connected.	

Returns

TRUE if all bytes written, otherwise FALSE

Flags: Run Last

Since: 3.0.0

Chapter 16

Audio OSC - OSC server

The Open Sound Control server and controllers. Every request is passed through the front controller. It delegates the request to the requested controller. Each controller has its own context path to match.

16.1 AgsOscServer

AgsOscServer — the OSC server

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_SERVER_GET_OBJ_MUTEX()
gboolean	ags_osc_server_test_flags ()
void	ags_osc_server_set_flags ()
void	ags_osc_server_unset_flags ()
void	ags_osc_server_add_connection ()
void	ags_osc_server_remove_connection ()
void	ags_osc_server_add_controller ()
void	ags_osc_server_remove_controller ()
void	ags_osc_server_add_default_controller ()
void	ags_osc_server_start ()
void	ags_osc_server_stop ()
gboolean	ags_osc_server_listen ()
void	ags_osc_server_dispatch ()
AgsOscServer *	ags_osc_server_new ()
#define	AGS_IS_OSC_SERVER()
#define	AGS_IS_OSC_SERVER_CLASS()
#define	AGS_OSC_SERVER()
#define	AGS_OSC_SERVER_CLASS()
#define	AGS_OSC_SERVER_GET_CLASS()
GType	ags_osc_server_flags_get_type ()
GType	ags_osc_server_get_type ()

Properties

gpointer	connection	Read / Write
gpointer	controller	Read / Write
char *	domain	Read / Write
AgsOscFrontController *	front-controller	Read / Write
char *	ip4	Read / Write
char *	ip6	Read / Write
guint	server-port	Read / Write

Signals

void	dispatch	Run Last
gboolean	listen	Run Last
void	start	Run Last
void	stop	Run Last

Types and Values

#define	AGS_OSC_SERVER_DEFAULT_MAX_ADDRESS_LENGTH
#define	AGS_OSC_SERVER_DEFAULT_SERVER_PORT
#define	AGS_OSC_SERVER_DEFAULT_DOMAIN
#define	AGS_OSC_SERVER_DEFAULT_INET4_ADDRESS
#define	AGS_OSC_SERVER_DEFAULT_INET6_ADDRESS
#define	AGS_OSC_SERVER_DEFAULT_BACKLOG
#define	AGS_OSC_SERVER_DEFAULT_MAX_CONNECTIONS
enum	AgsOscServerFlags
#define	AGS_TYPE_OSC_SERVER
struct	AgsOscServer
struct	AgsOscServerClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOscServerFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscServer
        &#x2570;&#x2500;&#x2500; AgsOscXmlrpcServer
```

Includes

```
#include <ags/audio/osc/ags_osc_server.h>
```

Description

AgsOscServer your osc server.

Functions

AGS_OSC_SERVER_GET_OBJ_MUTEX()

```
#define AGS_OSC_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsOscServer *) obj)->obj_mutex))
```

ags_osc_server_test_flags ()

```
gboolean
ags_osc_server_test_flags (AgsOscServer *osc_server,
                           guint flags);
```

Test *flags* to be set on *osc_server* .

Parameters

osc_server	the AgsOscServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_server_set_flags ()

```
void
ags_osc_server_set_flags (AgsOscServer *osc_server,
                          guint flags);
```

Set flags.

Parameters

osc_server	the AgsOscServer	
flags	the flags	

Since: 3.0.0

ags_osc_server_unset_flags ()

```
void
ags_osc_server_unset_flags (AgsOscServer *osc_server,
                            guint flags);
```

Unset flags.

Parameters

osc_server	the AgsOscServer	
flags	the flags	

Since: 3.0.0

ags_osc_server_add_connection ()

```
void
ags_osc_server_add_connection (AgsOscServer *osc_server,
                               GObject *osc_connection);
```

Add *osc_connection* to *osc_server* .

Parameters

osc_server	the AgsOscServer	
osc_connection	the AgsOscConnection	

Since: 3.0.0

ags_osc_server_remove_connection ()

```
void
ags_osc_server_remove_connection (AgsOscServer *osc_server,
                                  GObject *osc_connection);
```

Remove *osc_connection* from *osc_server* .

Parameters

osc_server	the AgsOscServer	
osc_connection	the AgsOscConnection	

Since: 3.0.0

ags_osc_server_add_controller ()

```
void
ags_osc_server_add_controller (AgsOscServer *osc_server,
                               GObject *osc_controller);
```

Add *osc_controller* to *osc_server* .

Parameters

osc_server	the AgsOscServer	
osc_controller	the AgsOscController	

Since: 3.0.0

ags_osc_server_remove_controller ()

```
void
ags_osc_server_remove_controller (AgsOscServer *osc_server,
                                  GObject *osc_controller);
```

Remove *osc_controller* from *osc_server* .

Parameters

osc_server	the AgsOscServer	
osc_controller	the AgsOscController	

Since: 3.0.0

ags_osc_server_add_default_controller ()

```
void
ags_osc_server_add_default_controller (AgsOscServer *osc_server);
```

Add default controlers to *osc_server* .

Parameters

osc_server	the AgsOscServer	

Since: 3.0.0

ags_osc_server_start ()

```
void
ags_osc_server_start (AgsOscServer *osc_server);
```

Start OSC server.

Parameters

osc_server	the AgsOscServer	

Since: 3.0.0

ags_osc_server_stop ()

```
void
ags_osc_server_stop (AgsOscServer *osc_server);
```

Stop OSC server.

Parameters

osc_server	the AgsOscServer	

Since: 3.0.0

ags_osc_server_listen ()

```
gboolean
ags_osc_server_listen (AgsOscServer *osc_server);
```

Listen as OSC server.

Parameters

osc_server	the AgsOscServer

Returns

TRUE as a new connection was initiated, otherwise FALSE

Since: 3.0.0

ags_osc_server_dispatch ()

```
void
ags_osc_server_dispatch (AgsOscServer *osc_server);
```

Dispatch the OSC server.

Parameters

osc_server	the AgsOscServer

Since: 3.0.0

ags_osc_server_new ()

```
AgsOscServer~*
ags_osc_server_new ();
```

Creates a new instance of AgsOscServer

Returns

the new AgsOscServer

Since: 3.0.0

AGS_IS_OSC_SERVER()

```
#define AGS_IS_OSC_SERVER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_SERVER))
```

AGS_IS_OSC_SERVER_CLASS()

```
#define AGS_IS_OSC_SERVER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_SERVER))
```

AGS_OSC_SERVER()

```
#define AGS_OSC_SERVER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_SERVER, AgsOscServer))
```

AGS_OSC_SERVER_CLASS()

```
#define AGS_OSC_SERVER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_SERVER, AgsOscServerClass))
```

AGS_OSC_SERVER_GET_CLASS()

```
#define AGS_OSC_SERVER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_SERVER, AgsOscServerClass))
```

ags_osc_server_flags_get_type ()

```
GType
ags_osc_server_flags_get_type ();
```

ags_osc_server_get_type ()

```
GType
ags_osc_server_get_type (void);
```

Types and Values

AGS_OSC_SERVER_DEFAULT_MAX_ADDRESS_LENGTH

```
#define AGS_OSC_SERVER_DEFAULT_MAX_ADDRESS_LENGTH (2048)
```

AGS_OSC_SERVER_DEFAULT_SERVER_PORT

```
#define AGS_OSC_SERVER_DEFAULT_SERVER_PORT (9000)
```

AGS_OSC_SERVER_DEFAULT_DOMAIN

```
#define AGS_OSC_SERVER_DEFAULT_DOMAIN "localhost"
```

AGS_OSC_SERVER_DEFAULT_INET4_ADDRESS

```
#define AGS_OSC_SERVER_DEFAULT_INET4_ADDRESS "127.0.0.1"
```

AGS_OSC_SERVER_DEFAULT_INET6_ADDRESS

```
#define AGS_OSC_SERVER_DEFAULT_INET6_ADDRESS "::1"
```

AGS_OSC_SERVER_DEFAULT_BACKLOG

```
#define AGS_OSC_SERVER_DEFAULT_BACKLOG (512)
```

AGS_OSC_SERVER_DEFAULT_MAX_CONNECTIONS

```
#define AGS_OSC_SERVER_DEFAULT_MAX_CONNECTIONS (8192)
```

enum AgsOscServerFlags

Enum values to control the behavior or indicate internal state of AgsOscServer by enable/disable as flags.

Members

AGS_OSC_SERVER_STARTED	the server was started
AGS_OSC_SERVER_RUNNING	the server is up and running
AGS_OSC_SERVER_TERMINATING	the server is closing connections and terminating
AGS_OSC_SERVER_INET4	use IPv4
AGS_OSC_SERVER_INET6	use IPv6

AGS_OSC_SERVER_UDP	run using UDP transport
AGS_OSC_SERVER_TCP	run using TCP transport
AGS_OSC_SERVER_UNIX	run on UNIX socket
AGS_OSC_SERVER_ANY_ADDRESS	listen on any address
AGS_OSC_SERVER_AUTO_START	start the server

AGS_TYPE_OSC_SERVER

```
#define AGS_TYPE_OSC_SERVER                (ags_osc_server_get_type ())
```

struct AgsOscServer

```
struct AgsOscServer;
```

struct AgsOscServerClass

```
struct AgsOscServerClass {
  GObjectClass gobject;

  void (*start)(AgsOscServer *osc_server);
  void (*stop)(AgsOscServer *osc_server);

  gboolean (*listen)(AgsOscServer *osc_server);

  void (*dispatch)(AgsOscServer *osc_server);
};
```

Property Details

The "connection" property

```
  "connection"               gpointer
```

The assigned AgsOscConnection providing default settings.

[transfer full]

Owner: AgsOscServer

Flags: Read / Write

Since: 3.0.0

The "controller" property

```
"controller"                gpointer
```

The assigned AgsOscController providing default settings.

[transfer full]

Owner: AgsOscServer

Flags: Read / Write

Since: 3.0.0

The "domain" property

```
"domain"                    char~*
```

The domain to use.

Owner: AgsOscServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "front-controller" property

```
"front-controller"          AgsOscFrontController~*
```

The assigned AgsOscFrontController.

Owner: AgsOscServer

Flags: Read / Write

Since: 3.0.0

The "ip4" property

```
"ip4"                       char~*
```

The IPv4 address as string of the server.

Owner: AgsOscServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ip6" property

```
  "ip6"                       char~*
```

The IPv6 address as string of the server.

Owner: AgsOscServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "server-port" property

```
  "server-port"               guint
```

The server port to use.

Owner: AgsOscServer

Flags: Read / Write

Default value: 9000

Since: 3.0.0

Signal Details

The "dispatch" signal

```
void
user_function (AgsOscServer *osc_server,
               gpointer      user_data)
```

The ::dispatch signal is emited during dispatch of server.

Parameters

osc_server	the AgsOscServer	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "listen" signal

```
gboolean
user_function (AgsOscServer *osc_server,
               gpointer      user_data)
```

The ::listen signal is emited during listen of server.

Parameters

osc_server	the AgsOscServer	
user_data	user data set when the signal handler was connected.	

Returns

TRUE as a new connection was initiated, otherwise FALSE

Flags: Run Last

Since: 3.0.0

The "start" signal

```
void
user_function (AgsOscServer *osc_server,
               gpointer      user_data)
```

The ::start signal is emited during start of server.

Parameters

osc_server	the AgsOscServer	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop" signal

```
void
user_function (AgsOscServer *osc_server,
               gpointer      user_data)
```

The ::stop signal is emited during stop of server.

Parameters

osc_server	the AgsOscServer	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

16.2 AgsOscConnection

AgsOscConnection — the OSC server side connection

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_CONNECTION_GET_OBJ_MUTEX()
gboolean	ags_osc_connection_test_flags ()
void	ags_osc_connection_set_flags ()
void	ags_osc_connection_unset_flags ()
gboolean	ags_osc_connection_timeout_expired ()
guchar *	ags_osc_connection_read_bytes ()
gint64	ags_osc_connection_write_response ()
void	ags_osc_connection_close ()
AgsOscConnection *	ags_osc_connection_new ()
#define	AGS_IS_OSC_CONNECTION()
#define	AGS_IS_OSC_CONNECTION_CLASS()
#define	AGS_OSC_CONNECTION()
#define	AGS_OSC_CONNECTION_CLASS()
#define	AGS_OSC_CONNECTION_GET_CLASS()
GType	ags_osc_connection_get_type ()

Properties

char *	ip4	Read / Write
char *	ip6	Read / Write
AgsOscServer *	osc-server	Read / Write

Signals

void	close	Run Last
gpointer	read-bytes	Run Last
gint64	write-response	Run Last

Types and Values

#define	AGS_OSC_CONNECTION_TIMEOUT_USEC
#define	AGS_OSC_CONNECTION_DEAD_LINE_USEC
#define	AGS_OSC_CONNECTION_CHUNK_SIZE
#define	AGS_OSC_CONNECTION_DEFAULT_CACHE_DATA_LENGTH
enum	AgsOscConnectionFlags
#define	AGS_TYPE_OSC_CONNECTION
struct	AgsOscConnection
struct	AgsOscConnectionClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscConnection
        &#x2570;&#x2500;&#x2500; AgsOscWebsocketConnection
```

Includes

```
#include <ags/audio/osc/ags_osc_connection.h>
```

Description

AgsOscConnection your OSC server side connection.

Functions

AGS_OSC_CONNECTION_GET_OBJ_MUTEX()

```
#define AGS_OSC_CONNECTION_GET_OBJ_MUTEX(obj) (&(((AgsOscConnection *) obj)->obj_mutex))
```

ags_osc_connection_test_flags ()

```
gboolean
ags_osc_connection_test_flags (AgsOscConnection *osc_connection,
                               guint flags);
```

Test *flags* to be set on *osc_connection* .

Parameters

osc_connection	the AgsOscConnection	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_connection_set_flags ()

```
void
ags_osc_connection_set_flags (AgsOscConnection *osc_connection,
                              guint flags);
```

Set flags.

Parameters

osc_connection	the AgsOscConnection	
flags	the flags	

Since: 3.0.0

ags_osc_connection_unset_flags ()

```
void
ags_osc_connection_unset_flags (AgsOscConnection *osc_connection,
                                guint flags);
```

Unset flags.

Parameters

osc_connection	the AgsOscConnection	
flags	the flags	

Since: 3.0.0

ags_osc_connection_timeout_expired ()

```
gboolean
ags_osc_connection_timeout_expired (struct timespec *start_time,
                                    struct timespec *timeout_delay);
```

Check *start_time* plus *timeout_delay* against current time.

Parameters

start_time	the start time timespec	
timeout_delay	the delay timespec	

Returns

TRUE if timeout expired, otherwise FALSE

Since: 3.0.0

ags_osc_connection_read_bytes ()

```
guchar~*
ags_osc_connection_read_bytes (AgsOscConnection *osc_connection,
                               guint *data_length);
```

Read bytes.

Parameters

osc_connection	the AgsOscConnection	
data_length	the return location of byte array's length	

Returns

byte array read or NULL if no data available

Since: 3.0.0

ags_osc_connection_write_response ()

```
gint64
ags_osc_connection_write_response (AgsOscConnection *osc_connection,
                                   GObject *osc_response);
```

Write response.

Parameters

osc_connection	the AgsOscConnection	
osc_response	the AgsOscResponse	

Returns

the count of bytes written

Since: 3.0.0

ags_osc_connection_close ()

```
void
ags_osc_connection_close (AgsOscConnection *osc_connection);
```

Close *osc_connection*.

Parameters

osc_connection	the AgsOscConnection	

Since: 3.0.0

ags_osc_connection_new ()

```
AgsOscConnection~*
ags_osc_connection_new (GObject *osc_server);
```

Creates a new instance of AgsOscConnection

Parameters

osc_server	the AgsOscServer

Returns

the new AgsOscConnection

Since: 3.0.0

AGS_IS_OSC_CONNECTION()

```
#define AGS_IS_OSC_CONNECTION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_CONNECTION))
```

AGS_IS_OSC_CONNECTION_CLASS()

```
#define AGS_IS_OSC_CONNECTION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_CONNECTION))
```

AGS_OSC_CONNECTION()

```
#define AGS_OSC_CONNECTION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_CONNECTION, AgsOscConnection))
```

AGS_OSC_CONNECTION_CLASS()

```
#define AGS_OSC_CONNECTION_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_CONNECTION, AgsOscConnectionClass))
```

AGS_OSC_CONNECTION_GET_CLASS()

```
#define AGS_OSC_CONNECTION_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_CONNECTION, AgsOscConnectionClass))
```

ags_osc_connection_get_type ()

```
GType
ags_osc_connection_get_type (void);
```

Types and Values

AGS_OSC_CONNECTION_TIMEOUT_USEC

```
#define AGS_OSC_CONNECTION_TIMEOUT_USEC (250)
```

AGS_OSC_CONNECTION_DEAD_LINE_USEC

```
#define AGS_OSC_CONNECTION_DEAD_LINE_USEC (60000000)
```

AGS_OSC_CONNECTION_CHUNK_SIZE

```
#define AGS_OSC_CONNECTION_CHUNK_SIZE (131072)
```

AGS_OSC_CONNECTION_DEFAULT_CACHE_DATA_LENGTH

```
#define AGS_OSC_CONNECTION_DEFAULT_CACHE_DATA_LENGTH (256)
```

enum AgsOscConnectionFlags

Enum values to configure OSC connection.

Members

AGS_OSC_CONNECTION_ACTIVE	is active
AGS_OSC_CONNECTION_INET4	IPv4 connection
AGS_OSC_CONNECTION_INET6	IPv6 connection

AGS_TYPE_OSC_CONNECTION

```
#define AGS_TYPE_OSC_CONNECTION                (ags_osc_connection_get_type ())
```

struct AgsOscConnection

```
struct AgsOscConnection;
```

struct AgsOscConnectionClass

```
struct AgsOscConnectionClass {
  GObjectClass gobject;

  guchar* (*read_bytes)(AgsOscConnection *osc_connection,
      guint *data_length);
  gint64 (*write_response)(AgsOscConnection *osc_connection,
         GObject *osc_response);

  void (*close)(AgsOscConnection *osc_connection);
};
```

Property Details

The "ip4" property

```
"ip4"                       char~*
```

The IPv4 address as string of the server connection.

Owner: AgsOscConnection

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ip6" property

```
"ip6"                       char~*
```

The IPv6 address as string of the server connection.

Owner: AgsOscConnection

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "osc-server" property

```
"osc-server"                AgsOscServer~*
```

The assigned AgsOscServer.

Owner: AgsOscConnection

Flags: Read / Write

Since: 3.0.0

Signal Details

The "close" signal

```
void
user_function (AgsOscConnection *osc_connection,
               gpointer          user_data)
```

The ::close signal is emited as closing the file descriptor.

Parameters

osc_connection	the AgsOscConnection	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "read-bytes" signal

```
gpointer
user_function (AgsOscConnection *osc_connection,
               gpointer          data_length,
               gpointer          user_data)
```

The ::read-bytes signal is emited while read bytes.

Parameters

osc_connection	the AgsOscConnection	
data_length	the return location of byte array's length	
user_data	user data set when the signal handler was connected.	

Returns

byte array read or NULL if no data available

Flags: Run Last

Since: 3.0.0

The "write-response" signal

```
gint64
user_function (AgsOscConnection *osc_connection,
               GObject          *osc_response,
               gpointer          user_data)
```

The ::write-response signal is emited while write response.

Parameters

osc_connection	the AgsOscConnection	
osc_response	the AgsOscResponse	
user_data	user data set when the signal handler was connected.	

Returns

the count bytes written

Flags: Run Last

Since: 3.0.0

16.3 AgsOscMessage

AgsOscMessage — the OSC server side message

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_MESSAGE_GET_OBJ_MUTEX()
gint	ags_osc_message_sort_func ()
AgsOscMessage *	ags_osc_message_new ()
#define	AGS_IS_OSC_MESSAGE()
#define	AGS_IS_OSC_MESSAGE_CLASS()
#define	AGS_OSC_MESSAGE()
#define	AGS_OSC_MESSAGE_CLASS()
#define	AGS_OSC_MESSAGE_GET_CLASS()
GType	ags_osc_message_get_type ()

Properties

gboolean	immediately	Read / Write
gpointer	message	Read / Write
guint	message-size	Read / Write
AgsOscConnection *	osc-connection	Read / Write
int	tv-fraction	Read / Write
int	tv-sec	Read / Write

Types and Values

#define	AGS_TYPE_OSC_MESSAGE
struct	AgsOscMessage
struct	AgsOscMessageClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsOscMessage
    &#x2570;&#x2500;&#x2500; AgsOscXmlrpcMessage
```

Includes

```
#include <ags/audio/osc/ags_osc_message.h>
```

Description

AgsOscMessage your OSC server side message.

Functions

AGS_OSC_MESSAGE_GET_OBJ_MUTEX()

```
#define AGS_OSC_MESSAGE_GET_OBJ_MUTEX(obj) (&(((AgsOscMessage *) obj)->obj_mutex))
```

ags_osc_message_sort_func ()

```
gint
ags_osc_message_sort_func (gconstpointer a,
                           gconstpointer b);
```

ags_osc_message_new ()

```
AgsOscMessage~*
ags_osc_message_new ();
```

Creates a new instance of AgsOscMessage

Returns

the new AgsOscMessage

Since: 3.0.0

AGS_IS_OSC_MESSAGE()

```
#define AGS_IS_OSC_MESSAGE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_MESSAGE))
```

AGS_IS_OSC_MESSAGE_CLASS()

```
#define AGS_IS_OSC_MESSAGE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_MESSAGE))
```

AGS_OSC_MESSAGE()

```
#define AGS_OSC_MESSAGE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_MESSAGE, AgsOscMessage))
```

AGS_OSC_MESSAGE_CLASS()

```
#define AGS_OSC_MESSAGE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_MESSAGE, AgsOscMessageClass))
```

AGS_OSC_MESSAGE_GET_CLASS()

```
#define AGS_OSC_MESSAGE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_MESSAGE, AgsOscMessageClass))
```

ags_osc_message_get_type ()

```
GType
ags_osc_message_get_type (void);
```

Types and Values

AGS_TYPE_OSC_MESSAGE

```
#define AGS_TYPE_OSC_MESSAGE                (ags_osc_message_get_type ())
```

struct AgsOscMessage

```
struct AgsOscMessage;
```

struct AgsOscMessageClass

```
struct AgsOscMessageClass {
  GObjectClass gobject;
};
```

Property Details

The "immediately" property

```
  "immediately"              gboolean
```

The immediately switch.

Owner: AgsOscMessage

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "message" property

```
  "message"                  gpointer
```

The message data.

Owner: AgsOscMessage

Flags: Read / Write

Since: 3.0.0

The "message-size" property

```
  "message-size"             guint
```

The size of message in bytes.

Owner: AgsOscMessage

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "osc-connection" property

```
"osc-connection"            AgsOscConnection~*
```

The assigned AgsOscConnection.

Owner: AgsOscMessage

Flags: Read / Write

Since: 3.0.0

The "tv-fraction" property

```
"tv-fraction"               int
```

The time value's fraction.

Owner: AgsOscMessage

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "tv-sec" property

```
"tv-sec"                    int
```

The time value in seconds.

Owner: AgsOscMessage

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

16.4 AgsOscResponse

AgsOscResponse — the OSC server side response

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_RESPONSE_GET_OBJ_MUTEX()
gboolean	ags_osc_response_test_flags ()
void	ags_osc_response_set_flags ()
void	ags_osc_response_unset_flags ()
AgsOscResponse *	ags_osc_response_new ()
#define	AGS_IS_OSC_RESPONSE()
#define	AGS_IS_OSC_RESPONSE_CLASS()
#define	AGS_OSC_RESPONSE()
#define	AGS_OSC_RESPONSE_CLASS()
#define	AGS_OSC_RESPONSE_GET_CLASS()
GType	ags_osc_response_get_type ()

Properties

char *	error-message	Read / Write
AgsOscMessage *	osc-message	Read / Write
gpointer	packet	Read / Write
guint	packet-size	Read / Write

Types and Values

#define	AGS_OSC_RESPONSE_ERROR_MESSAGE_SERVER_FAILURE
#define	AGS_OSC_RESPONSE_ERROR_MESSAGE_MALFORMED_REQUEST
#define	AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_PATH
#define	AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_ARGUMENT
#define	AGS_OSC_RESPONSE_ERROR_MESSAGE_MISSING_INDEX
#define	AGS_OSC_RESPONSE_ERROR_MESSAGE_CHUNK_SIZE_EXCEEDED
#define	AGS_OSC_RESPONSE_DEFAULT_CHUNK_SIZE
enum	AgsOscResponseFlags
#define	AGS_TYPE_OSC_RESPONSE
struct	AgsOscResponse
struct	AgsOscResponseClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsOscResponse
```

Includes

```
#include <ags/audio/osc/ags_osc_response.h>
```

Description

AgsOscResponse your OSC server side response.

Functions

AGS_OSC_RESPONSE_GET_OBJ_MUTEX()

```
#define AGS_OSC_RESPONSE_GET_OBJ_MUTEX(obj) (&(((AgsOscResponse *) obj)->obj_mutex))
```

ags_osc_response_test_flags ()

```
gboolean
ags_osc_response_test_flags (AgsOscResponse *osc_response,
                             guint flags);
```

Test *flags* to be set on *osc_response* .

Parameters

osc_response	the AgsOscResponse	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_response_set_flags ()

```
void
ags_osc_response_set_flags (AgsOscResponse *osc_response,
                            guint flags);
```

Set flags.

Parameters

osc_response	the AgsOscResponse	
flags	the flags	

Since: 3.0.0

ags_osc_response_unset_flags ()

```
void
ags_osc_response_unset_flags (AgsOscResponse *osc_response,
                              guint flags);
```

Unset flags.

Parameters

osc_response	the AgsOscResponse	
flags	the flags	

Since: 3.0.0

ags_osc_response_new ()

```
AgsOscResponse~*
ags_osc_response_new ();
```

Creates a new instance of AgsOscResponse

Returns

the new AgsOscResponse

Since: 3.0.0

AGS_IS_OSC_RESPONSE()

```
#define AGS_IS_OSC_RESPONSE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_RESPONSE))
```

AGS_IS_OSC_RESPONSE_CLASS()

```
#define AGS_IS_OSC_RESPONSE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_RESPONSE))
```

AGS_OSC_RESPONSE()

```
#define AGS_OSC_RESPONSE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_RESPONSE, AgsOscResponse))
```

AGS_OSC_RESPONSE_CLASS()

```
#define AGS_OSC_RESPONSE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_RESPONSE, AgsOscResponseClass))
```

AGS_OSC_RESPONSE_GET_CLASS()

```
#define AGS_OSC_RESPONSE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_RESPONSE, AgsOscResponseClass))
```

ags_osc_response_get_type ()

```
GType
ags_osc_response_get_type (void);
```

Types and Values

AGS_OSC_RESPONSE_ERROR_MESSAGE_SERVER_FAILURE

```
#define AGS_OSC_RESPONSE_ERROR_MESSAGE_SERVER_FAILURE "server failure"
```

AGS_OSC_RESPONSE_ERROR_MESSAGE_MALFORMED_REQUEST

```
#define AGS_OSC_RESPONSE_ERROR_MESSAGE_MALFORMED_REQUEST "malformed request"
```

AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_PATH

```
#define AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_PATH "unknown path"
```

AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_ARGUMENT

```
#define AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_ARGUMENT "unknown argument"
```

AGS_OSC_RESPONSE_ERROR_MESSAGE_MISSING_INDEX

```
#define AGS_OSC_RESPONSE_ERROR_MESSAGE_MISSING_INDEX "missing index"
```

AGS_OSC_RESPONSE_ERROR_MESSAGE_CHUNK_SIZE_EXCEEDED

```
#define AGS_OSC_RESPONSE_ERROR_MESSAGE_CHUNK_SIZE_EXCEEDED "chunk size exceeded"
```

AGS_OSC_RESPONSE_DEFAULT_CHUNK_SIZE

```
#define AGS_OSC_RESPONSE_DEFAULT_CHUNK_SIZE (131072)
```

enum AgsOscResponseFlags

Enum values to configure OSC response.

Members

AGS_OSC_RESPONSE_EMPTY	is empty
AGS_OSC_RESPONSE_OK	is OK
AGS_OSC_RESPONSE_ERROR	has error

AGS_TYPE_OSC_RESPONSE

```
#define AGS_TYPE_OSC_RESPONSE                (ags_osc_response_get_type ())
```

struct AgsOscResponse

```
struct AgsOscResponse;
```

struct AgsOscResponseClass

```
struct AgsOscResponseClass {
  GObjectClass gobject;
};
```

Property Details

The "error-message" property

```
  "error-message"            char~*
```

The error message.

Owner: AgsOscResponse

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "osc-message" property

```
  "osc-message"              AgsOscMessage~*
```

The assigned AgsOscMessage

Owner: AgsOscResponse

Flags: Read / Write

Since: 3.0.0

The "packet" property

```
  "packet"                   gpointer
```

The response packet.

Owner: AgsOscResponse

Flags: Read / Write

Since: 3.0.0

The "packet-size" property

```
"packet-size"              guint
```

The response packet's size.

Owner: AgsOscResponse

Flags: Read / Write

Default value: 0

Since: 3.0.0

16.5 AgsOscController

AgsOscController — base OSC controller

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_CONTROLLER_GET_OBJ_MUTEX()
AgsOscController *	ags_osc_controller_new ()
#define	AGS_IS_OSC_CONTROLLER()
#define	AGS_IS_OSC_CONTROLLER_CLASS()
#define	AGS_OSC_CONTROLLER()
#define	AGS_OSC_CONTROLLER_CLASS()
#define	AGS_OSC_CONTROLLER_GET_CLASS()
GType	ags_osc_controller_get_type ()

Properties

char *	context-path	Read / Write
AgsOscServer *	osc-server	Read / Write

Types and Values

#define	AGS_TYPE_OSC_CONTROLLER
struct	AgsOscController
struct	AgsOscControllerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsOscController
    &#x251c;&#x2500;&#x2500; AgsOscActionController
    &#x251c;&#x2500;&#x2500; AgsOscConfigController
    &#x251c;&#x2500;&#x2500; AgsOscExportController
    &#x251c;&#x2500;&#x2500; AgsOscFrontController
```

```
        &#x251c;&#x2500;&#x2500; AgsOscInfoController
        &#x251c;&#x2500;&#x2500; AgsOscMeterController
        &#x251c;&#x2500;&#x2500; AgsOscNodeController
        &#x251c;&#x2500;&#x2500; AgsOscRenewController
        &#x2570;&#x2500;&#x2500; AgsOscStatusController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_controller.h>
```

Description

The AgsOscController is a base object to implement OSC controllers.

Functions

AGS_OSC_CONTROLLER_GET_OBJ_MUTEX()

```
#define AGS_OSC_CONTROLLER_GET_OBJ_MUTEX(obj) (&(((AgsOscController *) obj)->obj_mutex))
```

ags_osc_controller_new ()

```
AgsOscController~*
ags_osc_controller_new ();
```

Instantiate new AgsOscController

Returns

the AgsOscController

Since: 3.0.0

AGS_IS_OSC_CONTROLLER()

```
#define AGS_IS_OSC_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_CONTROLLER))
```

AGS_IS_OSC_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_CONTROLLER))
```

AGS_OSC_CONTROLLER()

```
#define AGS_OSC_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_CONTROLLER, AgsOscController))
```

AGS_OSC_CONTROLLER_CLASS()

```
#define AGS_OSC_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_CONTROLLER, AgsOscControllerClass))
```

AGS_OSC_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_CONTROLLER, AgsOscControllerClass))
```

ags_osc_controller_get_type ()

```
GType
ags_osc_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_CONTROLLER

```
#define AGS_TYPE_OSC_CONTROLLER                 (ags_osc_controller_get_type())
```

struct AgsOscController

```
struct AgsOscController;
```

struct AgsOscControllerClass

```
struct AgsOscControllerClass {
  GObjectClass gobject;
};
```

Property Details

The "context-path" property

```
  "context-path"              char~*
```

The context path provided.

Owner: AgsOscController

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "osc-server" property

```
"osc-server"                AgsOscServer~*
```

The assigned AgsOscServer

Owner: AgsOscController

Flags: Read / Write

Since: 3.0.0

16.6 AgsOscActionController

AgsOscActionController — OSC action controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_action_controller_run_action ()
AgsOscActionController *	ags_osc_action_controller_new ()
#define	AGS_IS_OSC_ACTION_CONTROLLER()
#define	AGS_IS_OSC_ACTION_CONTROLLER_CLASS()
#define	AGS_OSC_ACTION_CONTROLLER()
#define	AGS_OSC_ACTION_CONTROLLER_CLASS()
#define	AGS_OSC_ACTION_CONTROLLER_GET_CLASS()
GType	ags_osc_action_controller_get_type ()

Signals

gpointer	run-action	Run Last

Types and Values

#define	AGS_TYPE_OSC_ACTION_CONTROLLER
struct	AgsOscActionController
struct	AgsOscActionControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscActionController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_action_controller.h>
```

Description

The AgsOscActionController implements the OSC action controller.

Functions

ags_osc_action_controller_run_action ()

```
gpointer
ags_osc_action_controller_run_action (AgsOscActionController *osc_action_controller,
                                      AgsOscConnection *osc_connection,
                                      guchar *message,
                                      guint message_size);
```

Run action.

Parameters

osc_action_controller	the AgsOscActionController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_action_controller_new ()

```
AgsOscActionController~*
ags_osc_action_controller_new ();
```

Instantiate new AgsOscActionController

Returns

the AgsOscActionController

Since: 3.0.0

AGS_IS_OSC_ACTION_CONTROLLER()

```
#define AGS_IS_OSC_ACTION_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_ACTION_CONTROLLER))
```

AGS_IS_OSC_ACTION_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_ACTION_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_ACTION_CONTROLLER))
```

AGS_OSC_ACTION_CONTROLLER()

```
#define AGS_OSC_ACTION_CONTROLLER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_ACTION_CONTROLLER, AgsOscActionController))
```

AGS_OSC_ACTION_CONTROLLER_CLASS()

```
#define AGS_OSC_ACTION_CONTROLLER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_ACTION_CONTROLLER, AgsOscActionControllerClass))
```

AGS_OSC_ACTION_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_ACTION_CONTROLLER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_ACTION_CONTROLLER, AgsOscActionControllerClass))
```

ags_osc_action_controller_get_type ()

```
GType
ags_osc_action_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_ACTION_CONTROLLER

```
#define AGS_TYPE_OSC_ACTION_CONTROLLER                (ags_osc_action_controller_get_type() ↩
    )
```

struct AgsOscActionController

```
struct AgsOscActionController;
```

struct AgsOscActionControllerClass

```
struct AgsOscActionControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*run_action)(AgsOscActionController *osc_action_controller,
       AgsOscConnection *osc_connection,
       guchar *message, guint message_size);
};
```

Signal Details

The "run-action" signal

```
gpointer
user_function (AgsOscActionController *osc_action_controller,
               GObject                *osc_connection,
               gpointer                message,
               guint                   message_size,
               gpointer                user_data)
```

The ::run-action signal is emited during get data of action controller.

Parameters

osc_action_controller	the AgsOscActionController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

16.7 AgsOscConfigController

AgsOscConfigController — OSC config controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_config_controller_apply_config ()
AgsOscConfigController *	ags_osc_config_controller_new ()
#define	AGS_IS_OSC_CONFIG_CONTROLLER()
#define	AGS_IS_OSC_CONFIG_CONTROLLER_CLASS()
#define	AGS_OSC_CONFIG_CONTROLLER()
#define	AGS_OSC_CONFIG_CONTROLLER_CLASS()
#define	AGS_OSC_CONFIG_CONTROLLER_GET_CLASS()
GType	ags_osc_config_controller_get_type ()

Signals

gpointer	apply-config	Run Last

Types and Values

#define	AGS_TYPE_OSC_CONFIG_CONTROLLER
struct	AgsOscConfigController
struct	AgsOscConfigControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscConfigController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_config_controller.h>
```

Description

The AgsOscConfigController implements the OSC config controller.

Functions

ags_osc_config_controller_apply_config ()

```
gpointer
ags_osc_config_controller_apply_config
                                (AgsOscConfigController *osc_config_controller,
                                 AgsOscConnection *osc_connection,
                                 guchar *message,
                                 guint message_size);
```

Apply config.

Parameters

osc_config_controller	the AgsOscConfigController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_config_controller_new ()

```
AgsOscConfigController~*
ags_osc_config_controller_new ();
```

Instantiate new AgsOscConfigController

Returns

the AgsOscConfigController

Since: 3.0.0

AGS_IS_OSC_CONFIG_CONTROLLER()

```
#define AGS_IS_OSC_CONFIG_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_CONFIG_CONTROLLER))
```

AGS_IS_OSC_CONFIG_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_CONFIG_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_CONFIG_CONTROLLER))
```

AGS_OSC_CONFIG_CONTROLLER()

```
#define AGS_OSC_CONFIG_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_CONFIG_CONTROLLER, AgsOscConfigController))
```

AGS_OSC_CONFIG_CONTROLLER_CLASS()

```
#define AGS_OSC_CONFIG_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_CONFIG_CONTROLLER, AgsOscConfigControllerClass))
```

AGS_OSC_CONFIG_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_CONFIG_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_CONFIG_CONTROLLER, AgsOscConfigControllerClass))
```

ags_osc_config_controller_get_type ()

```
GType
ags_osc_config_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_CONFIG_CONTROLLER

```
#define AGS_TYPE_OSC_CONFIG_CONTROLLER                 (ags_osc_config_controller_get_type() ↩
    )
```

struct AgsOscConfigController

```
struct AgsOscConfigController;
```

struct AgsOscConfigControllerClass

```
struct AgsOscConfigControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*apply_config)(AgsOscConfigController *osc_config_controller,
          AgsOscConnection *osc_connection,
          guchar *message, guint message_size);
};
```

Signal Details

The "apply-config" signal

```
gpointer
user_function (AgsOscConfigController *osc_config_controller,
               GObject                *osc_connection,
               gpointer                message,
               guint                   message_size,
               gpointer                user_data)
```

The ::apply-config signal is emited during get data of config controller.

Parameters

osc_config_controller	the AgsOscConfigController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

16.8 AgsOscExportController

AgsOscExportController — OSC export controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_export_controller_do_export ()
AgsOscExportController *	ags_osc_export_controller_new ()
#define	AGS_IS_OSC_EXPORT_CONTROLLER()
#define	AGS_IS_OSC_EXPORT_CONTROLLER_CLASS()
#define	AGS_OSC_EXPORT_CONTROLLER()
#define	AGS_OSC_EXPORT_CONTROLLER_CLASS()
#define	AGS_OSC_EXPORT_CONTROLLER_GET_CLASS()
GType	ags_osc_export_controller_get_type ()

Signals

gpointer	do-export	Run Last

Types and Values

#define	AGS_TYPE_OSC_EXPORT_CONTROLLER
struct	AgsOscExportController
struct	AgsOscExportControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscExportController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_export_controller.h>
```

Description

The AgsOscExportController implements the OSC export controller.

Functions

ags_osc_export_controller_do_export ()

```
gpointer
ags_osc_export_controller_do_export (AgsOscExportController *osc_export_controller,
                                     AgsOscConnection *osc_connection,
                                     guchar *message,
                                     guint message_size);
```

Run export.

Parameters

osc_export_controller	the AgsOscExportController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_export_controller_new ()

```
AgsOscExportController~*
ags_osc_export_controller_new ();
```

Instantiate new AgsOscExportController

Returns

the AgsOscExportController

Since: 3.0.0

AGS_IS_OSC_EXPORT_CONTROLLER()

```
#define AGS_IS_OSC_EXPORT_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_EXPORT_CONTROLLER))
```

AGS_IS_OSC_EXPORT_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_EXPORT_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_EXPORT_CONTROLLER))
```

AGS_OSC_EXPORT_CONTROLLER()

```
#define AGS_OSC_EXPORT_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_EXPORT_CONTROLLER, AgsOscExportController))
```

AGS_OSC_EXPORT_CONTROLLER_CLASS()

```
#define AGS_OSC_EXPORT_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_EXPORT_CONTROLLER, AgsOscExportControllerClass))
```

AGS_OSC_EXPORT_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_EXPORT_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_EXPORT_CONTROLLER, AgsOscExportControllerClass))
```

ags_osc_export_controller_get_type ()

```
GType
ags_osc_export_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_EXPORT_CONTROLLER

```
#define AGS_TYPE_OSC_EXPORT_CONTROLLER                (ags_osc_export_controller_get_type() ↩
    )
```

struct AgsOscExportController

```
struct AgsOscExportController;
```

struct AgsOscExportControllerClass

```
struct AgsOscExportControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*do_export)(AgsOscExportController *osc_export_controller,
      AgsOscConnection *osc_connection,
      guchar *message, guint message_size);
};
```

Signal Details

The "do-export" signal

```
gpointer
user_function (AgsOscExportController *osc_export_controller,
               GObject                *osc_connection,
               gpointer                message,
               guint                   message_size,
               gpointer                user_data)
```

The ::do-export signal is emited during get data of export controller.

Parameters

osc_export_controller	the AgsOscExportController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

16.9 AgsOscFrontController

AgsOscFrontController — OSC front controller

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_osc_front_controller_test_flags ()
void	ags_osc_front_controller_set_flags ()
void	ags_osc_front_controller_unset_flags ()
void	ags_osc_front_controller_add_message ()
void	ags_osc_front_controller_remove_message ()
void	ags_osc_front_controller_start_delegate ()
void	ags_osc_front_controller_stop_delegate ()
gpointer	ags_osc_front_controller_do_request ()
AgsOscFrontController *	ags_osc_front_controller_new ()
#define	AGS_IS_OSC_FRONT_CONTROLLER()
#define	AGS_IS_OSC_FRONT_CONTROLLER_CLASS()
#define	AGS_OSC_FRONT_CONTROLLER()
#define	AGS_OSC_FRONT_CONTROLLER_CLASS()
#define	AGS_OSC_FRONT_CONTROLLER_GET_CLASS()
GType	ags_osc_front_controller_flags_get_type ()
GType	ags_osc_front_controller_get_type ()

Signals

gpointer	do-request	Run Last
void	start-delegate	Run Last
void	stop-delegate	Run Last

Types and Values

enum	AgsOscFrontControllerFlags
#define	AGS_TYPE_OSC_FRONT_CONTROLLER
struct	AgsOscFrontController
struct	AgsOscFrontControllerClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOscFrontControllerFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscFrontController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_front_controller.h>
```

Description

The AgsOscFrontController implements the OSC front controller.

Functions

ags_osc_front_controller_test_flags ()

```
gboolean
ags_osc_front_controller_test_flags (AgsOscFrontController *osc_front_controller,
                                     guint flags);
```

Test *flags* to be set on *osc_front_controller* .

Parameters

osc_front_controller	the AgsOscFrontController	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_front_controller_set_flags ()

```
void
ags_osc_front_controller_set_flags (AgsOscFrontController *osc_front_controller,
                                    guint flags);
```

Set flags.

Parameters

osc_front_controller	the AgsOscFrontController	
flags	the flags	

Since: 3.0.0

ags_osc_front_controller_unset_flags ()

```
void
ags_osc_front_controller_unset_flags (AgsOscFrontController *osc_front_controller,
                                      guint flags);
```

Unset flags.

Parameters

osc_front_controller	the AgsOscFrontController	
flags	the flags	

Since: 3.0.0

ags_osc_front_controller_add_message ()

```
void
ags_osc_front_controller_add_message (AgsOscFrontController *osc_front_controller,
                                      GObject *message);
```

Add *message* to *osc_front_controller* .

Parameters

osc_front_controller	the AgsOscFrontController	
message	the AgsOscMessage	

Since: 3.0.0

ags_osc_front_controller_remove_message ()

```
void
ags_osc_front_controller_remove_message
                                (AgsOscFrontController *osc_front_controller,
                                 GObject *message);
```

Remove *message* from *osc_front_controller* .

Parameters

osc_front_controller	the AgsOscFrontController	
message	the AgsOscMessage	

Since: 3.0.0

ags_osc_front_controller_start_delegate ()

```
void
ags_osc_front_controller_start_delegate
                                (AgsOscFrontController *osc_front_controller);
```

Start delegating.

Parameters

osc_front_controller	the AgsOscFrontController

Since: 3.0.0

ags_osc_front_controller_stop_delegate ()

```
void
ags_osc_front_controller_stop_delegate
                                (AgsOscFrontController *osc_front_controller);
```

Stop delegating.

Parameters

osc_front_controller	the AgsOscFrontController

Since: 3.0.0

ags_osc_front_controller_do_request ()

```
gpointer
ags_osc_front_controller_do_request (AgsOscFrontController *osc_front_controller,
                                     AgsOscConnection *osc_connection,
                                     guchar *packet,
                                     gsize packet_size);
```

ags_osc_front_controller_new ()

```
AgsOscFrontController~*
ags_osc_front_controller_new ();
```

Instantiate new AgsOscFrontController

Returns

the AgsOscFrontController

Since: 3.0.0

AGS_IS_OSC_FRONT_CONTROLLER()

```
#define AGS_IS_OSC_FRONT_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_FRONT_CONTROLLER))
```

AGS_IS_OSC_FRONT_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_FRONT_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_FRONT_CONTROLLER))
```

AGS_OSC_FRONT_CONTROLLER()

```
#define AGS_OSC_FRONT_CONTROLLER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_FRONT_CONTROLLER, AgsOscFrontController))
```

AGS_OSC_FRONT_CONTROLLER_CLASS()

```
#define AGS_OSC_FRONT_CONTROLLER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_FRONT_CONTROLLER, AgsOscFrontControllerClass))
```

AGS_OSC_FRONT_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_FRONT_CONTROLLER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_FRONT_CONTROLLER, AgsOscFrontControllerClass))
```

ags_osc_front_controller_flags_get_type ()

```
GType
ags_osc_front_controller_flags_get_type
                                ();
```

ags_osc_front_controller_get_type ()

```
GType
ags_osc_front_controller_get_type ();
```

Types and Values

enum AgsOscFrontControllerFlags

Enum values to control the behavior or indicate internal state of AgsOscFrontController by enable/disable as flags.

Members

AGS_OSC_FRONT_CONTROLLER_DELEGATE_STARTED		
AGS_OSC_FRONT_CONTROLLER_DELEGATE_RUNNING		
AGS_OSC_FRONT_CONTROLLER_DELEGATE_TERMINATING		

AGS_TYPE_OSC_FRONT_CONTROLLER

```
#define AGS_TYPE_OSC_FRONT_CONTROLLER                (ags_osc_front_controller_get_type())
```

struct AgsOscFrontController

```
struct AgsOscFrontController;
```

struct AgsOscFrontControllerClass

```
struct AgsOscFrontControllerClass {
  AgsOscControllerClass osc_controller;

  void (*start_delegate)(AgsOscFrontController *osc_front_controller);
  void (*stop_delegate)(AgsOscFrontController *osc_front_controller);

  gpointer (*do_request)(AgsOscFrontController *osc_front_controller,
       AgsOscConnection *osc_connection,
       guchar *packet, gsize packet_size);
};
```

Signal Details

The "do-request" signal

```
gpointer
user_function (AgsOscFrontController *osc_front_controller,
               GObject               *osc_connection,
               gpointer               packet,
               gint64                 packet_size,
               gpointer               user_data)
```

The ::do-request signal is emited during do request of front controller.

Parameters

osc_front_controller	the AgsOscFrontController	
osc_connection	the AgsOscConnection	
packet	the packet received	
packet_size	the packet size	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "start-delegate" signal

```
void
user_function (AgsOscFrontController *osc_front_controller,
               gpointer               user_data)
```

The ::start-delegate signal is emited during start of delegating messages.

Parameters

osc_front_controller	the AgsOscFrontController	

user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop-delegate" signal

```
void
user_function (AgsOscFrontController *osc_front_controller,
               gpointer               user_data)
```

The ::stop-delegate signal is emited during stop of delegating messages.

Parameters

osc_front_controller	the AgsOscFrontController	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

16.10 AgsOscInfoController

AgsOscInfoController — OSC info controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_info_controller_get_info ()
AgsOscInfoController *	ags_osc_info_controller_new ()
#define	AGS_IS_OSC_INFO_CONTROLLER()
#define	AGS_IS_OSC_INFO_CONTROLLER_CLASS()
#define	AGS_OSC_INFO_CONTROLLER()
#define	AGS_OSC_INFO_CONTROLLER_CLASS()
#define	AGS_OSC_INFO_CONTROLLER_GET_CLASS()
GType	ags_osc_info_controller_get_type ()

Signals

gpointer	get-info	Run Last

Types and Values

#define	AGS_TYPE_OSC_INFO_CONTROLLER
struct	AgsOscInfoController
struct	AgsOscInfoControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscInfoController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_info_controller.h>
```

Description

The AgsOscInfoController implements the OSC info controller.

Functions

ags_osc_info_controller_get_info ()

```
gpointer
ags_osc_info_controller_get_info (AgsOscInfoController *osc_info_controller,
                                  AgsOscConnection *osc_connection,
                                  guchar *message,
                                  guint message_size);
```

Get info.

Parameters

osc_info_controller	the AgsOscInfoController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_info_controller_new ()

```
AgsOscInfoController~*
ags_osc_info_controller_new ();
```

Instantiate new AgsOscInfoController

Returns

the AgsOscInfoController

Since: 3.0.0

AGS_IS_OSC_INFO_CONTROLLER()

```
#define AGS_IS_OSC_INFO_CONTROLLER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_INFO_CONTROLLER))
```

AGS_IS_OSC_INFO_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_INFO_CONTROLLER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_INFO_CONTROLLER))
```

AGS_OSC_INFO_CONTROLLER()

```
#define AGS_OSC_INFO_CONTROLLER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_INFO_CONTROLLER, AgsOscInfoController))
```

AGS_OSC_INFO_CONTROLLER_CLASS()

```
#define AGS_OSC_INFO_CONTROLLER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_INFO_CONTROLLER, AgsOscInfoControllerClass))
```

AGS_OSC_INFO_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_INFO_CONTROLLER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_INFO_CONTROLLER, AgsOscInfoControllerClass))
```

ags_osc_info_controller_get_type ()

```
GType
ags_osc_info_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_INFO_CONTROLLER

```
#define AGS_TYPE_OSC_INFO_CONTROLLER                (ags_osc_info_controller_get_type())
```

struct AgsOscInfoController

```
struct AgsOscInfoController;
```

struct AgsOscInfoControllerClass

```
struct AgsOscInfoControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*get_info)(AgsOscInfoController *osc_info_controller,
           AgsOscConnection *osc_connection,
           guchar *message, guint message_size);
};
```

Signal Details

The "get-info" signal

```
gpointer
user_function (AgsOscInfoController *osc_info_controller,
               GObject              *osc_connection,
               gpointer              message,
               guint                 message_size,
               gpointer              user_data)
```

The ::get-info signal is emited during get info of info controller.

Parameters

osc_info_controller	the AgsOscInfoController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

16.11 AgsOscMeterController

AgsOscMeterController — base osc_meter_controller

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_OSC_METER_CONTROLLER_MONITOR()
gboolean	ags_osc_meter_controller_test_flags ()
void	ags_osc_meter_controller_set_flags ()
void	ags_osc_meter_controller_unset_flags ()
AgsOscMeterControllerMonitor *	ags_osc_meter_controller_monitor_alloc ()
void	ags_osc_meter_controller_monitor_free ()
void	ags_osc_meter_controller_monitor_ref ()
void	ags_osc_meter_controller_monitor_unref ()
GList *	ags_osc_meter_controller_monitor_find_path ()
GList *	ags_osc_meter_controller_monitor_find_port ()
void	ags_osc_meter_controller_add_monitor ()
void	ags_osc_meter_controller_remove_monitor ()
gboolean	ags_osc_meter_controller_contains_monitor ()
void	ags_osc_meter_controller_start_monitor ()
void	ags_osc_meter_controller_stop_monitor ()
gpointer	ags_osc_meter_controller_monitor_meter ()
AgsOscMeterController *	ags_osc_meter_controller_new ()
#define	AGS_IS_OSC_METER_CONTROLLER()
#define	AGS_IS_OSC_METER_CONTROLLER_CLASS()
#define	AGS_OSC_METER_CONTROLLER()
#define	AGS_OSC_METER_CONTROLLER_CLASS()
#define	AGS_OSC_METER_CONTROLLER_GET_CLASS()
GType	ags_osc_meter_controller_get_type ()

Signals

gpointer	monitor-meter	Run Last
void	start-monitor	Run Last
void	stop-monitor	Run Last

Types and Values

#define	AGS_OSC_METER_CONTROLLER_DEFAULT_MONITOR_TIMEOUT
enum	AgsOscMeterControllerFlags
struct	AgsOscMeterControllerMonitor
#define	AGS_TYPE_OSC_METER_CONTROLLER
struct	AgsOscMeterController
struct	AgsOscMeterControllerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsOscController
    &#x2570;&#x2500;&#x2500; AgsOscMeterController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_meter_controller.h>
```

Description

The AgsOscMeterController is a base object to implement osc_meter_controllers.

Functions

AGS_OSC_METER_CONTROLLER_MONITOR()

```
#define AGS_OSC_METER_CONTROLLER_MONITOR(ptr) ((AgsOscMeterControllerMonitor *)(ptr))
```

ags_osc_meter_controller_test_flags ()

```
gboolean
ags_osc_meter_controller_test_flags (AgsOscMeterController *osc_meter_controller,
                                     guint flags);
```

Test *flags* to be set on *osc_meter_controller* .

Parameters

osc_meter_controller	the AgsOscMeterController	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_meter_controller_set_flags ()

```
void
ags_osc_meter_controller_set_flags (AgsOscMeterController *osc_meter_controller,
                                    guint flags);
```

Set flags.

Parameters

osc_meter_controller	the AgsOscMeterController	
flags	the flags	

Since: 3.0.0

ags_osc_meter_controller_unset_flags ()

```
void
ags_osc_meter_controller_unset_flags (AgsOscMeterController *osc_meter_controller,
                                      guint flags);
```

Unset flags.

Parameters

osc_meter_controller	the AgsOscMeterController	
flags	the flags	

Since: 3.0.0

ags_osc_meter_controller_monitor_alloc ()

```
AgsOscMeterControllerMonitor~*
ags_osc_meter_controller_monitor_alloc
                                ();
```

Allocate AgsOscMeterControllerMonitor.

Returns

the newly allocate AgsOscMeterControllerMonitor.

[type gpointer][transfer none]

Since: 3.0.0

ags_osc_meter_controller_monitor_free ()

```
void
ags_osc_meter_controller_monitor_free (AgsOscMeterControllerMonitor *monitor);
```

Free *monitor* .

Parameters

monitor	the AgsOscMeterControllerMonitor.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_osc_meter_controller_monitor_ref ()

```
void
ags_osc_meter_controller_monitor_ref (AgsOscMeterControllerMonitor *monitor);
```

Increase reference count of *monitor* .

Parameters

monitor	the AgsOscMeterControllerMonitor.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_osc_meter_controller_monitor_unref ()

```
void
ags_osc_meter_controller_monitor_unref
                                (AgsOscMeterControllerMonitor *monitor);
```

Decrease reference count of *monitor* . If ref count is less or equal 0 *monitor* is freed.

Parameters

monitor	the AgsOscMeterControllerMonitor.	*[type gpointer][transfer none]*

Since: 3.0.0

ags_osc_meter_controller_monitor_find_path ()

```
GList~*
ags_osc_meter_controller_monitor_find_path
                                (GList *monitor,
                                 gchar *path);
```

Find *path* in *monitor* .

Parameters

monitor	the GList containing AgsOscMeterController-Monitor.	*[element-type gpointer][transfer none]*
path	the path as string	

Returns

the next matching GList containing AgsOscMeterControllerMonitor.

[element-type gpointer][transfer none]

Since: 3.0.0

ags_osc_meter_controller_monitor_find_port ()

```
GList~*
ags_osc_meter_controller_monitor_find_port
                                (GList *monitor,
                                 AgsPort *port);
```

Find *port* in *monitor* .

Parameters

monitor	the GList containing AgsOscMeterController-Monitor.	*[element-type gpointer][transfer none]*
port	the AgsPort	

Returns

the next matching GList containing AgsOscMeterControllerMonitor.

[element-type gpointer][transfer none]

Since: 3.0.0

ags_osc_meter_controller_add_monitor ()

```
void
ags_osc_meter_controller_add_monitor (AgsOscMeterController *osc_meter_controller,
                                      AgsOscMeterControllerMonitor *monitor);
```

ags_osc_meter_controller_remove_monitor ()

```
void
ags_osc_meter_controller_remove_monitor
                                (AgsOscMeterController *osc_meter_controller,
                                 AgsOscMeterControllerMonitor *monitor);
```

ags_osc_meter_controller_contains_monitor ()

```
gboolean
ags_osc_meter_controller_contains_monitor
                                (AgsOscMeterController *osc_meter_controller,
                                 AgsOscConnection *osc_connection,
                                 AgsPort *port);
```

ags_osc_meter_controller_start_monitor ()

```
void
ags_osc_meter_controller_start_monitor
                                (AgsOscMeterController *osc_meter_controller);
```

Start monitoring.

Parameters

osc_meter_controller	the AgsOscMeterController

Since: 3.0.0

ags_osc_meter_controller_stop_monitor ()

```
void
ags_osc_meter_controller_stop_monitor (AgsOscMeterController *osc_meter_controller);
```

Stop monitoring.

Parameters

osc_meter_controller	the AgsOscMeterController

Since: 3.0.0

ags_osc_meter_controller_monitor_meter ()

```
gpointer
ags_osc_meter_controller_monitor_meter
                                (AgsOscMeterController *osc_meter_controller,
                                 AgsOscConnection *osc_connection,
                                 guchar *message,
                                 guint message_size);
```

Get meter.

Parameters

osc_meter_controller	the AgsOscMeterController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_meter_controller_new ()

```
AgsOscMeterController~*
ags_osc_meter_controller_new ();
```

Instantiate new AgsOscMeterController

Returns

the AgsOscMeterController

Since: 3.0.0

AGS_IS_OSC_METER_CONTROLLER()

```
#define AGS_IS_OSC_METER_CONTROLLER(obj)            (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_METER_CONTROLLER))
```

AGS_IS_OSC_METER_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_METER_CONTROLLER_CLASS(class)    (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_METER_CONTROLLER))
```

AGS_OSC_METER_CONTROLLER()

```
#define AGS_OSC_METER_CONTROLLER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_METER_CONTROLLER, AgsOscMeterController))
```

AGS_OSC_METER_CONTROLLER_CLASS()

```
#define AGS_OSC_METER_CONTROLLER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_METER_CONTROLLER, AgsOscMeterControllerClass))
```

AGS_OSC_METER_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_METER_CONTROLLER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_METER_CONTROLLER, AgsOscMeterControllerClass))
```

ags_osc_meter_controller_get_type ()

```
GType
ags_osc_meter_controller_get_type ();
```

Types and Values

AGS_OSC_METER_CONTROLLER_DEFAULT_MONITOR_TIMEOUT

```
#define AGS_OSC_METER_CONTROLLER_DEFAULT_MONITOR_TIMEOUT (1.0 / 30.0)
```

enum AgsOscMeterControllerFlags

Members

AGS_OSC_METER_CONTROLLER_MONITOR_STARTED	
AGS_OSC_METER_CONTROLLER_MONITOR_RUNNING	
AGS_OSC_METER_CONTROLLER_MONITOR_TERMINATING	

struct AgsOscMeterControllerMonitor

```
struct AgsOscMeterControllerMonitor {
  volatile gint ref_count;

  AgsOscConnection *osc_connection;

  gchar *path;
  AgsPort *port;
};
```

AGS_TYPE_OSC_METER_CONTROLLER

```
#define AGS_TYPE_OSC_METER_CONTROLLER                (ags_osc_meter_controller_get_type())
```

struct AgsOscMeterController

```
struct AgsOscMeterController;
```

struct AgsOscMeterControllerClass

```
struct AgsOscMeterControllerClass {
  AgsOscControllerClass osc_controller;

  void (*start_monitor)(AgsOscMeterController *osc_meter_controller);
  void (*stop_monitor)(AgsOscMeterController *osc_meter_controller);

  gpointer (*monitor_meter)(AgsOscMeterController *osc_meter_controller,
          AgsOscConnection *osc_connection,
          guchar *message, guint message_size);
};
```

Signal Details

The "monitor-meter" signal

```
gpointer
user_function (AgsOscMeterController *osc_meter_controller,
               GObject               *osc_connection,
               gpointer               message,
               guint                  message_size,
               gpointer               user_data)
```

The ::monitor-meter signal is emited during get data of meter controller.

Parameters

osc_meter_controller	the AgsOscMeterController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

The "start-monitor" signal

```
void
user_function (AgsOscMeterController *osc_meter_controller,
               gpointer               user_data)
```

The ::start-monitor signal is emited during start of monitoring meter.

Parameters

osc_meter_controller	the AgsOscMeterController	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop-monitor" signal

```
void
user_function (AgsOscMeterController *osc_meter_controller,
               gpointer               user_data)
```

The ::stop-monitor signal is emited during stop of monitoring meter.

Parameters

osc_meter_controller	the AgsOscMeterController	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

16.12 AgsOscNodeController

AgsOscNodeController — OSC node controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_node_controller_get_data ()
AgsOscNodeController *	ags_osc_node_controller_new ()
#define	AGS_IS_OSC_NODE_CONTROLLER()
#define	AGS_IS_OSC_NODE_CONTROLLER_CLASS()
#define	AGS_OSC_NODE_CONTROLLER()
#define	AGS_OSC_NODE_CONTROLLER_CLASS()
#define	AGS_OSC_NODE_CONTROLLER_GET_CLASS()
GType	ags_osc_node_controller_get_type ()

Signals

gpointer	get-data	Run Last

Types and Values

#define	AGS_OSC_NODE_CONTROLLER_CONTEXT_PATH_LENGTH
#define	AGS_TYPE_OSC_NODE_CONTROLLER
struct	AgsOscNodeController
struct	AgsOscNodeControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscNodeController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_node_controller.h>
```

Description

The AgsOscNodeController implements the OSC node controller.

Functions

ags_osc_node_controller_get_data ()

```
gpointer
ags_osc_node_controller_get_data (AgsOscNodeController *osc_node_controller,
                                  AgsOscConnection *osc_connection,
                                  guchar *message,
                                  guint message_size);
```

Get node.

Parameters

osc_node_controller	the AgsOscNodeController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_node_controller_new ()

```
AgsOscNodeController~*
ags_osc_node_controller_new ();
```

Instantiate new AgsOscNodeController

Returns

the AgsOscNodeController

Since: 3.0.0

AGS_IS_OSC_NODE_CONTROLLER()

```
#define AGS_IS_OSC_NODE_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_NODE_CONTROLLER))
```

AGS_IS_OSC_NODE_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_NODE_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_NODE_CONTROLLER))
```

AGS_OSC_NODE_CONTROLLER()

```
#define AGS_OSC_NODE_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_NODE_CONTROLLER, AgsOscNodeController))
```

AGS_OSC_NODE_CONTROLLER_CLASS()

```
#define AGS_OSC_NODE_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_NODE_CONTROLLER, AgsOscNodeControllerClass))
```

AGS_OSC_NODE_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_NODE_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_NODE_CONTROLLER, AgsOscNodeControllerClass))
```

ags_osc_node_controller_get_type ()

```
GType
ags_osc_node_controller_get_type ();
```

Types and Values

AGS_OSC_NODE_CONTROLLER_CONTEXT_PATH_LENGTH

```
#define AGS_OSC_NODE_CONTROLLER_CONTEXT_PATH_LENGTH (8)
```

AGS_TYPE_OSC_NODE_CONTROLLER

```
#define AGS_TYPE_OSC_NODE_CONTROLLER                (ags_osc_node_controller_get_type())
```

struct AgsOscNodeController

```
struct AgsOscNodeController;
```

struct AgsOscNodeControllerClass

```
struct AgsOscNodeControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*get_data)(AgsOscNodeController *osc_node_controller,
           AgsOscConnection *osc_connection,
           guchar *message, guint message_size);
};
```

Signal Details

The "get-data" signal

```
gpointer
user_function (AgsOscNodeController *osc_node_controller,
               GObject              *osc_connection,
               gpointer              message,
               guint                 message_size,
               gpointer              user_data)
```

The ::get-data signal is emited during get data of node controller.

Parameters

osc_node_controller	the AgsOscNodeController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

16.13 AgsOscPluginController

AgsOscPluginController — provide abstract OSC controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_plugin_controller_do_request ()
#define	AGS_IS_OSC_PLUGIN_CONTROLLER()
#define	AGS_IS_OSC_PLUGIN_CONTROLLER_INTERFACE()
#define	AGS_OSC_PLUGIN_CONTROLLER()
#define	AGS_OSC_PLUGIN_CONTROLLER_GET_INTERFACE()
#define	AGS_OSC_PLUGIN_CONTROLLER_INTERFACE()
GType	ags_osc_plugin_controller_get_type ()

Types and Values

#define	AGS_TYPE_OSC_PLUGIN_CONTROLLER
	AgsOscPluginController
struct	AgsOscPluginControllerInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsOscPluginController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_plugin_controller.h>
```

Description

The AgsOscPluginController provides you an abstraction of OSC controller implementation.

Functions

ags_osc_plugin_controller_do_request ()

```
gpointer
ags_osc_plugin_controller_do_request (AgsOscPluginController *osc_plugin_controller,
                                      AgsOscConnection *osc_connection,
                                      guchar *message,
                                      guint message_size);
```

Do request.

Parameters

osc_plugin_controller	the AgsOscPluginController	
osc_connection	the AgsOscConnection	
message	the OSC message	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

AGS_IS_OSC_PLUGIN_CONTROLLER()

```
#define AGS_IS_OSC_PLUGIN_CONTROLLER(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj) ↩
    , AGS_TYPE_OSC_PLUGIN_CONTROLLER))
```

AGS_IS_OSC_PLUGIN_CONTROLLER_INTERFACE()

```
#define AGS_IS_OSC_PLUGIN_CONTROLLER_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable) ↩
    , AGS_TYPE_OSC_PLUGIN_CONTROLLER))
```

AGS_OSC_PLUGIN_CONTROLLER()

```
#define AGS_OSC_PLUGIN_CONTROLLER(obj)                       (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_OSC_PLUGIN_CONTROLLER, AgsOscPluginController))
```

AGS_OSC_PLUGIN_CONTROLLER_GET_INTERFACE()

```
#define AGS_OSC_PLUGIN_CONTROLLER_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE(( ↩
    obj), AGS_TYPE_OSC_PLUGIN_CONTROLLER, AgsOscPluginControllerInterface))
```

AGS_OSC_PLUGIN_CONTROLLER_INTERFACE()

```
#define AGS_OSC_PLUGIN_CONTROLLER_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable) ↩
    , AGS_TYPE_OSC_PLUGIN_CONTROLLER, AgsOscPluginControllerInterface))
```

ags_osc_plugin_controller_get_type ()

```
GType
ags_osc_plugin_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_PLUGIN_CONTROLLER

```
#define AGS_TYPE_OSC_PLUGIN_CONTROLLER                       ( ↩
    ags_osc_plugin_controller_get_type())
```

AgsOscPluginController

```
typedef struct _AgsOscPluginController AgsOscPluginController;
```

struct AgsOscPluginControllerInterface

```
struct AgsOscPluginControllerInterface {
  GTypeInterface ginterface;

  gpointer (*do_request)(AgsOscPluginController *osc_plugin_controller,
       AgsOscConnection *osc_connection,
       guchar *message, guint message_size);
};
```

16.14 AgsOscRenewController

AgsOscRenewController — OSC renew controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_renew_controller_set_data ()
AgsOscRenewController *	ags_osc_renew_controller_new ()
#define	AGS_IS_OSC_RENEW_CONTROLLER()
#define	AGS_IS_OSC_RENEW_CONTROLLER_CLASS()
#define	AGS_OSC_RENEW_CONTROLLER()
#define	AGS_OSC_RENEW_CONTROLLER_CLASS()
#define	AGS_OSC_RENEW_CONTROLLER_GET_CLASS()
GType	ags_osc_renew_controller_get_type ()

Signals

gpointer	set-data	Run Last

Types and Values

#define	AGS_OSC_RENEW_CONTROLLER_CONTEXT_PATH_LENGTH
#define	AGS_TYPE_OSC_RENEW_CONTROLLER
struct	AgsOscRenewController
struct	AgsOscRenewControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscRenewController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_renew_controller.h>
```

Description

The AgsOscRenewController implements the OSC renew controller.

Functions

ags_osc_renew_controller_set_data ()

```
gpointer
ags_osc_renew_controller_set_data (AgsOscRenewController *osc_renew_controller,
                                   AgsOscConnection *osc_connection,
                                   guchar *message,
                                   guint message_size);
```

Renew data.

Parameters

osc_renew_controller	the AgsOscRenewController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_renew_controller_new ()

```
AgsOscRenewController~*
ags_osc_renew_controller_new ();
```

Instantiate new AgsOscRenewController

Returns

the AgsOscRenewController

Since: 3.0.0

AGS_IS_OSC_RENEW_CONTROLLER()

```
#define AGS_IS_OSC_RENEW_CONTROLLER(obj)            (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_RENEW_CONTROLLER))
```

AGS_IS_OSC_RENEW_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_RENEW_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_RENEW_CONTROLLER))
```

AGS_OSC_RENEW_CONTROLLER()

```
#define AGS_OSC_RENEW_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_RENEW_CONTROLLER, AgsOscRenewController))
```

AGS_OSC_RENEW_CONTROLLER_CLASS()

```
#define AGS_OSC_RENEW_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_RENEW_CONTROLLER, AgsOscRenewControllerClass))
```

AGS_OSC_RENEW_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_RENEW_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_RENEW_CONTROLLER, AgsOscRenewControllerClass))
```

ags_osc_renew_controller_get_type ()

```
GType
ags_osc_renew_controller_get_type ();
```

Types and Values

AGS_OSC_RENEW_CONTROLLER_CONTEXT_PATH_LENGTH

```
#define AGS_OSC_RENEW_CONTROLLER_CONTEXT_PATH_LENGTH (8)
```

AGS_TYPE_OSC_RENEW_CONTROLLER

```
#define AGS_TYPE_OSC_RENEW_CONTROLLER                 (ags_osc_renew_controller_get_type())
```

struct AgsOscRenewController

```
struct AgsOscRenewController;
```

struct AgsOscRenewControllerClass

```
struct AgsOscRenewControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*set_data)(AgsOscRenewController *osc_renew_controller,
           AgsOscConnection *osc_connection,
           guchar *message, guint message_size);
};
```

Signal Details

The "set-data" signal

```
gpointer
user_function (AgsOscRenewController *osc_renew_controller,
               GObject               *osc_connection,
               gpointer               message,
               guint                  message_size,
               gpointer               user_data)
```

The ::set-data signal is emited during get data of renew controller.

Parameters

osc_renew_controller	the AgsOscRenewController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

16.15 AgsOscStatusController

AgsOscStatusController — OSC status controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_osc_status_controller_get_status ()
AgsOscStatusController *	ags_osc_status_controller_new ()
#define	AGS_IS_OSC_STATUS_CONTROLLER()
#define	AGS_IS_OSC_STATUS_CONTROLLER_CLASS()
#define	AGS_OSC_STATUS_CONTROLLER()
#define	AGS_OSC_STATUS_CONTROLLER_CLASS()
#define	AGS_OSC_STATUS_CONTROLLER_GET_CLASS()
GType	ags_osc_status_controller_get_type ()

Signals

gpointer	get-status	Run Last

Types and Values

#define	AGS_TYPE_OSC_STATUS_CONTROLLER
struct	AgsOscStatusController
struct	AgsOscStatusControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscController
        &#x2570;&#x2500;&#x2500; AgsOscStatusController
```

Includes

```
#include <ags/audio/osc/controller/ags_osc_status_controller.h>
```

Description

The AgsOscStatusController implements the OSC status controller.

Functions

ags_osc_status_controller_get_status ()

```
gpointer
ags_osc_status_controller_get_status (AgsOscStatusController *osc_status_controller,
                                      AgsOscConnection *osc_connection,
                                      guchar *message,
                                      guint message_size);
```

Get status.

Parameters

osc_status_controller	the AgsOscStatusController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	

Returns

the GList containing AgsOscResponse

Since: 3.0.0

ags_osc_status_controller_new ()

```
AgsOscStatusController~*
ags_osc_status_controller_new ();
```

Instantiate new AgsOscStatusController

Returns

the AgsOscStatusController

Since: 3.0.0

AGS_IS_OSC_STATUS_CONTROLLER()

```
#define AGS_IS_OSC_STATUS_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_STATUS_CONTROLLER))
```

AGS_IS_OSC_STATUS_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_STATUS_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_STATUS_CONTROLLER))
```

AGS_OSC_STATUS_CONTROLLER()

```
#define AGS_OSC_STATUS_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_STATUS_CONTROLLER, AgsOscStatusController))
```

AGS_OSC_STATUS_CONTROLLER_CLASS()

```
#define AGS_OSC_STATUS_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_STATUS_CONTROLLER, AgsOscStatusControllerClass))
```

AGS_OSC_STATUS_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_STATUS_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_STATUS_CONTROLLER, AgsOscStatusControllerClass))
```

ags_osc_status_controller_get_type ()

```
GType
ags_osc_status_controller_get_type ();
```

Types and Values

AGS_TYPE_OSC_STATUS_CONTROLLER

```
#define AGS_TYPE_OSC_STATUS_CONTROLLER                 (ags_osc_status_controller_get_type() ↩
    )
```

struct AgsOscStatusController

```
struct AgsOscStatusController;
```

struct AgsOscStatusControllerClass

```
struct AgsOscStatusControllerClass {
  AgsOscControllerClass osc_controller;

  gpointer (*get_status)(AgsOscStatusController *osc_status_controller,
       AgsOscConnection *osc_connection,
       guchar *message, guint message_size);
};
```

Signal Details

The "get-status" signal

```
gpointer
user_function (AgsOscStatusController *osc_status_controller,
               GObject                *osc_connection,
               gpointer                message,
               guint                   message_size,
               gpointer                user_data)
```

The ::get-status signal is emited during get status of status controller.

Parameters

osc_status_controller	the AgsOscStatusController	
osc_connection	the AgsOscConnection	
message	the message received	
message_size	the message size	
user_data	user data set when the signal handler was connected.	

Returns

the AgsOscResponse

Flags: Run Last

Since: 3.0.0

Chapter 17

Audio OSC XMLRPC - OSC over XMLRPC server

The Open Sound Control server throught XMLRPC server.

17.1 AgsOscXmlrpcServer

AgsOscXmlrpcServer — the OSC XMLRPC server

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_osc_xmlrpc_server_add_websocket_handler ()
void	ags_osc_xmlrpc_server_add_default_controller ()
AgsOscXmlrpcServer *	ags_osc_xmlrpc_server_new ()
#define	AGS_IS_OSC_XMLRPC_SERVER()
#define	AGS_IS_OSC_XMLRPC_SERVER_CLASS()
#define	AGS_OSC_XMLRPC_SERVER()
#define	AGS_OSC_XMLRPC_SERVER_CLASS()
#define	AGS_OSC_XMLRPC_SERVER_GET_CLASS()
GType	ags_osc_xmlrpc_server_get_type ()

Properties

AgsController *	osc-xmlrpc-controller	Read / Write
AgsServer *	xmlrpc-server	Read / Write

Types and Values

#define	AGS_OSC_XMLRPC_SERVER_DEFAULT_CHUNK_SIZE
#define	AGS_TYPE_OSC_XMLRPC_SERVER
struct	AgsOscXmlrpcServer
struct	AgsOscXmlrpcServerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscServer
        &#x2570;&#x2500;&#x2500; AgsOscXmlrpcServer
```

Includes

```
#include <ags/audio/osc/ags_osc_xmlrpc_server.h>
```

Description

AgsOscXmlrpcServer your OSC XMLRPC server.

Functions

ags_osc_xmlrpc_server_add_websocket_handler ()

```
void
ags_osc_xmlrpc_server_add_websocket_handler
                                (AgsOscXmlrpcServer *osc_xmlrpc_server,
                                 char *path,
                                 char *origin,
                                 char **protocols,
                                 SoupServerWebsocketCallback callback,
                                 gpointer user_data,
                                 GDestroyNotify destroy);
```

Add websocket handler to soup server.

Parameters

osc_xmlrpc_server	the AgsOscXmlrpcServer	
path	the path	
origin	the origin	
protocols	the protocls as string vector	
callback	the callback	
user_data	user data	
destroy	destroy notify function	

Since: 3.0.0

ags_osc_xmlrpc_server_add_default_controller ()

```
void
ags_osc_xmlrpc_server_add_default_controller
                                (AgsOscXmlrpcServer *osc_xmlrpc_server);
```

ags_osc_xmlrpc_server_new ()

```
AgsOscXmlrpcServer~*
ags_osc_xmlrpc_server_new ();
```

Creates a new instance of AgsOscXmlrpcServer

Returns

the new AgsOscXmlrpcServer

Since: 3.0.0

AGS_IS_OSC_XMLRPC_SERVER()

```
#define AGS_IS_OSC_XMLRPC_SERVER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_SERVER))
```

AGS_IS_OSC_XMLRPC_SERVER_CLASS()

```
#define AGS_IS_OSC_XMLRPC_SERVER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_XMLRPC_SERVER))
```

AGS_OSC_XMLRPC_SERVER()

```
#define AGS_OSC_XMLRPC_SERVER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_SERVER, AgsOscXmlrpcServer))
```

AGS_OSC_XMLRPC_SERVER_CLASS()

```
#define AGS_OSC_XMLRPC_SERVER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_XMLRPC_SERVER, AgsOscXmlrpcServerClass))
```

AGS_OSC_XMLRPC_SERVER_GET_CLASS()

```
#define AGS_OSC_XMLRPC_SERVER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_SERVER, AgsOscXmlrpcServerClass))
```

ags_osc_xmlrpc_server_get_type ()

```
GType
ags_osc_xmlrpc_server_get_type (void);
```

Types and Values

AGS_OSC_XMLRPC_SERVER_DEFAULT_CHUNK_SIZE

```
#define AGS_OSC_XMLRPC_SERVER_DEFAULT_CHUNK_SIZE (16384)
```

AGS_TYPE_OSC_XMLRPC_SERVER

```
#define AGS_TYPE_OSC_XMLRPC_SERVER                (ags_osc_xmlrpc_server_get_type ())
```

struct AgsOscXmlrpcServer

```
struct AgsOscXmlrpcServer;
```

struct AgsOscXmlrpcServerClass

```
struct AgsOscXmlrpcServerClass {
  AgsOscServerClass osc_server;
};
```

Property Details

The "osc-xmlrpc-controller" property

```
  "osc-xmlrpc-controller"    AgsController~*
```

The assigned AgsOscXmlrpcController.

Owner: AgsOscXmlrpcServer

Flags: Read / Write

Since: 3.0.0

The "xmlrpc-server" property

```
  "xmlrpc-server"            AgsServer~*
```

The assigned AgsServer.

Owner: AgsOscXmlrpcServer

Flags: Read / Write

Since: 3.0.0

17.2 AgsOscWebsocketConnection

AgsOscWebsocketConnection — the OSC server side WebSockets connection

Stability Level

Stable, unless otherwise indicated

Functions

GList *	ags_osc_websocket_connection_find_resource_id ()
AgsOscWebsocketConnection *	ags_osc_websocket_connection_new ()
#define	AGS_IS_OSC_WEBSOCKET_CONNECTION()
#define	AGS_IS_OSC_WEBSOCKET_CONNECTION_CLASS()
#define	AGS_OSC_WEBSOCKET_CONNECTION()
#define	AGS_OSC_WEBSOCKET_CONNECTION_CLASS()
#define	AGS_OSC_WEBSOCKET_CONNECTION_GET_CLASS()
GType	ags_osc_websocket_connection_get_type ()

Properties

char *	login	Read / Write
char *	resource-id	Read / Write
AgsSecurityContext *	security-context	Read / Write
char *	security-token	Read / Write
SoupWebsocketConnection *	websocket-connection	Read / Write

Types and Values

#define	AGS_TYPE_OSC_WEBSOCKET_CONNECTION
struct	AgsOscWebsocketConnection
struct	AgsOscWebsocketConnectionClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscConnection
        &#x2570;&#x2500;&#x2500; AgsOscWebsocketConnection
```

Includes

```
#include <ags/audio/osc/ags_osc_websocket_connection.h>
```

Description

AgsOscWebsocketConnection your OSC server side websocket_connection.

Functions

ags_osc_websocket_connection_find_resource_id ()

```
GList~*
ags_osc_websocket_connection_find_resource_id
                                (GList *osc_websocket_connection,
                                 gchar *resource_id);
```

Find *resource_id* in *osc_websocket_connection* .

Parameters

osc_websocket_connection	the GList containing AgsOscWebsocketConnection.	*[element-type AgsAudio.OscWebsocketConnection][transfer none]*
resource_id	the resource id	

Returns

the next matchine GList or NULL.

[element-type AgsAudio.OscWebsocketConnection][transfer none]

Since: 3.0.0

ags_osc_websocket_connection_new ()

```
AgsOscWebsocketConnection~*
ags_osc_websocket_connection_new (GObject *osc_server);
```

Creates a new instance of AgsOscWebsocketConnection

Parameters

osc_server	the AgsOscServer	

Returns

the new AgsOscWebsocketConnection

Since: 3.0.0

AGS_IS_OSC_WEBSOCKET_CONNECTION()

```
#define AGS_IS_OSC_WEBSOCKET_CONNECTION(obj)            (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_OSC_WEBSOCKET_CONNECTION))
```

AGS_IS_OSC_WEBSOCKET_CONNECTION_CLASS()

```
#define AGS_IS_OSC_WEBSOCKET_CONNECTION_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_OSC_WEBSOCKET_CONNECTION))
```

AGS_OSC_WEBSOCKET_CONNECTION()

```
#define AGS_OSC_WEBSOCKET_CONNECTION(obj)               (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_OSC_WEBSOCKET_CONNECTION, AgsOscWebsocketConnection))
```

AGS_OSC_WEBSOCKET_CONNECTION_CLASS()

```
#define AGS_OSC_WEBSOCKET_CONNECTION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_WEBSOCKET_CONNECTION, AgsOscWebsocketConnectionClass))
```

AGS_OSC_WEBSOCKET_CONNECTION_GET_CLASS()

```
#define AGS_OSC_WEBSOCKET_CONNECTION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
    AGS_TYPE_OSC_WEBSOCKET_CONNECTION, AgsOscWebsocketConnectionClass))
```

ags_osc_websocket_connection_get_type ()

```
GType
ags_osc_websocket_connection_get_type (void);
```

Types and Values

AGS_TYPE_OSC_WEBSOCKET_CONNECTION

```
#define AGS_TYPE_OSC_WEBSOCKET_CONNECTION                ( ↩
    ags_osc_websocket_connection_get_type ())
```

struct AgsOscWebsocketConnection

```
struct AgsOscWebsocketConnection;
```

struct AgsOscWebsocketConnectionClass

```
struct AgsOscWebsocketConnectionClass {
  AgsOscConnectionClass connection;
};
```

Property Details

The "login" property

```
  "login"                    char~*
```

The current login.

Owner: AgsOscWebsocketConnection

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "resource-id" property

```
  "resource-id"              char~*
```

The resource ID from a redirect.

Owner: AgsOscWebsocketConnection

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "security-context" property

```
"security-context"          AgsSecurityContext~*
```

The assigned AgsSecurityContext.

Owner: AgsOscWebsocketConnection

Flags: Read / Write

Since: 3.0.0

The "security-token" property

```
"security-token"            char~*
```

The current security token.

Owner: AgsOscWebsocketConnection

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "websocket-connection" property

```
"websocket-connection"      SoupWebsocketConnection~*
```

The assigned SoupMessage.

Owner: AgsOscWebsocketConnection

Flags: Read / Write

Since: 3.0.0

17.3 AgsOscXmlrpcMessage

AgsOscXmlrpcMessage — the OSC server side XMLRPC message

Stability Level

Stable, unless otherwise indicated

Functions

AgsOscXmlrpcMessage *	ags_osc_xmlrpc_message_new ()
#define	AGS_IS_OSC_XMLRPC_MESSAGE()
#define	AGS_IS_OSC_XMLRPC_MESSAGE_CLASS()
#define	AGS_OSC_XMLRPC_MESSAGE()
#define	AGS_OSC_XMLRPC_MESSAGE_CLASS()
#define	AGS_OSC_XMLRPC_MESSAGE_GET_CLASS()
GType	ags_osc_xmlrpc_message_get_type ()

Properties

gpointer	query	Read / Write
SoupServerMessage *	server-msg	Read / Write

Types and Values

#define	AGS_TYPE_OSC_XMLRPC_MESSAGE
struct	AgsOscXmlrpcMessage
struct	AgsOscXmlrpcMessageClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsOscMessage
        &#x2570;&#x2500;&#x2500; AgsOscXmlrpcMessage
```

Includes

```
#include <ags/audio/osc/ags_osc_xmlrpc_message.h>
```

Description

AgsOscXmlrpcMessage your OSC server side XMLRPC message.

Functions

ags_osc_xmlrpc_message_new ()

```
AgsOscXmlrpcMessage~*
ags_osc_xmlrpc_message_new ();
```

Creates a new instance of AgsOscXmlrpcMessage

Returns

the new AgsOscXmlrpcMessage

Since: 3.0.0

AGS_IS_OSC_XMLRPC_MESSAGE()

```
#define AGS_IS_OSC_XMLRPC_MESSAGE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_MESSAGE))
```

AGS_IS_OSC_XMLRPC_MESSAGE_CLASS()

```
#define AGS_IS_OSC_XMLRPC_MESSAGE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_XMLRPC_MESSAGE))
```

AGS_OSC_XMLRPC_MESSAGE()

```
#define AGS_OSC_XMLRPC_MESSAGE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_MESSAGE, AgsOscXmlrpcMessage))
```

AGS_OSC_XMLRPC_MESSAGE_CLASS()

```
#define AGS_OSC_XMLRPC_MESSAGE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OSC_XMLRPC_MESSAGE, AgsOscXmlrpcMessageClass))
```

AGS_OSC_XMLRPC_MESSAGE_GET_CLASS()

```
#define AGS_OSC_XMLRPC_MESSAGE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_MESSAGE, AgsOscXmlrpcMessageClass))
```

ags_osc_xmlrpc_message_get_type ()

```
GType
ags_osc_xmlrpc_message_get_type (void);
```

Types and Values

AGS_TYPE_OSC_XMLRPC_MESSAGE

```
#define AGS_TYPE_OSC_XMLRPC_MESSAGE                 (ags_osc_xmlrpc_message_get_type ())
```

struct AgsOscXmlrpcMessage

```
struct AgsOscXmlrpcMessage;
```

struct AgsOscXmlrpcMessageClass

```
struct AgsOscXmlrpcMessageClass {
  AgsOscMessageClass osc_message;
};
```

Property Details

The "query" property

```
  "query"                     gpointer
```

The current query.

Owner: AgsOscXmlrpcMessage

Flags: Read / Write

Since: 3.0.0

The "server-msg" property

```
"server-msg"                SoupServerMessage~*
```

The assigned SoupMessage.

Owner: AgsOscXmlrpcMessage

Flags: Read / Write

Since: 4.0.0

17.4 AgsOscXmlrpcController

AgsOscXmlrpcController — handle OSC XMLRPC requests

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_osc_xmlrpc_controller_test_flags ()
void	ags_osc_xmlrpc_controller_set_flags ()
void	ags_osc_xmlrpc_controller_unset_flags ()
void	ags_osc_xmlrpc_controller_add_message ()
void	ags_osc_xmlrpc_controller_remove_message ()
void	ags_osc_xmlrpc_controller_start_delegate ()
void	ags_osc_xmlrpc_controller_stop_delegate ()
AgsOscXmlrpcController *	ags_osc_xmlrpc_controller_new ()
#define	AGS_IS_OSC_XMLRPC_CONTROLLER()
#define	AGS_IS_OSC_XMLRPC_CONTROLLER_CLASS()
#define	AGS_OSC_XMLRPC_CONTROLLER()
#define	AGS_OSC_XMLRPC_CONTROLLER_CLASS()
#define	AGS_OSC_XMLRPC_CONTROLLER_GET_CLASS()
GType	ags_osc_xmlrpc_controller_flags_get_type ()
GType	ags_osc_xmlrpc_controller_get_type ()

Properties

AgsOscXmlrpcServer *	osc-xmlrpc-server	Read / Write

Signals

void	start-delegate	Run Last
void	stop-delegate	Run Last

Types and Values

#define	AGS_OSC_XMLRPC_CONTROLLER_DEFAULT_DELEGATE_TIMEOUT
enum	AgsOscXmlrpcControllerFlags
#define	AGS_TYPE_OSC_XMLRPC_CONTROLLER
struct	AgsOscXmlrpcController
struct	AgsOscXmlrpcControllerClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsOscXmlrpcControllerFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsController
        &#x2570;&#x2500;&#x2500; AgsOscXmlrpcController
```

Implemented Interfaces

AgsOscXmlrpcController implements AgsPluginController.

Includes

```
#include <ags/audio/osc/controller/ags_osc_xmlrpc_controller.h>
```

Description

The AgsOscXmlrpcController is an object to handle XMLRPC requests.

Functions

ags_osc_xmlrpc_controller_test_flags ()

```
gboolean
ags_osc_xmlrpc_controller_test_flags (AgsOscXmlrpcController *osc_xmlrpc_controller,
                                      guint flags);
```

Test *flags* to be set on *osc_xmlrpc_controller* .

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_osc_xmlrpc_controller_set_flags ()

```
void
ags_osc_xmlrpc_controller_set_flags (AgsOscXmlrpcController *osc_xmlrpc_controller,
                                     guint flags);
```

Set flags.

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
flags	the flags	

Since: 3.0.0

ags_osc_xmlrpc_controller_unset_flags ()

```
void
ags_osc_xmlrpc_controller_unset_flags (AgsOscXmlrpcController *osc_xmlrpc_controller,
                                       guint flags);
```

Unset flags.

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
flags	the flags	

Since: 3.0.0

ags_osc_xmlrpc_controller_add_message ()

```
void
ags_osc_xmlrpc_controller_add_message (AgsOscXmlrpcController *osc_xmlrpc_controller,
                                       GObject *message);
```

Add *message* to *osc_xmlrpc_controller* .

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
message	the AgsOscMessage	

Since: 3.0.0

ags_osc_xmlrpc_controller_remove_message ()

```
void
ags_osc_xmlrpc_controller_remove_message
                                (AgsOscXmlrpcController *osc_xmlrpc_controller,
                                 GObject *message);
```

Remove *message* from *osc_xmlrpc_controller* .

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
message	the AgsOscMessage	

Since: 3.0.0

ags_osc_xmlrpc_controller_start_delegate ()

```
void
ags_osc_xmlrpc_controller_start_delegate
                                (AgsOscXmlrpcController *osc_xmlrpc_controller);
```

Start delegating.

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	

Since: 3.0.0

ags_osc_xmlrpc_controller_stop_delegate ()

```
void
ags_osc_xmlrpc_controller_stop_delegate
                                (AgsOscXmlrpcController *osc_xmlrpc_controller);
```

Stop delegating.

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	

Since: 3.0.0

ags_osc_xmlrpc_controller_new ()

```
AgsOscXmlrpcController~*
ags_osc_xmlrpc_controller_new ();
```

Instantiate new AgsOscXmlrpcController

Returns

the AgsOscXmlrpcController

Since: 3.0.0

AGS_IS_OSC_XMLRPC_CONTROLLER()

```
#define AGS_IS_OSC_XMLRPC_CONTROLLER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_CONTROLLER))
```

AGS_IS_OSC_XMLRPC_CONTROLLER_CLASS()

```
#define AGS_IS_OSC_XMLRPC_CONTROLLER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_OSC_XMLRPC_CONTROLLER))
```

AGS_OSC_XMLRPC_CONTROLLER()

```
#define AGS_OSC_XMLRPC_CONTROLLER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OSC_XMLRPC_CONTROLLER, AgsOscXmlrpcController))
```

AGS_OSC_XMLRPC_CONTROLLER_CLASS()

```
#define AGS_OSC_XMLRPC_CONTROLLER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_OSC_XMLRPC_CONTROLLER, AgsOscXmlrpcControllerClass))
```

AGS_OSC_XMLRPC_CONTROLLER_GET_CLASS()

```
#define AGS_OSC_XMLRPC_CONTROLLER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_OSC_XMLRPC_CONTROLLER, AgsOscXmlrpcControllerClass))
```

ags_osc_xmlrpc_controller_flags_get_type ()

```
GType
ags_osc_xmlrpc_controller_flags_get_type
                                ();
```

ags_osc_xmlrpc_controller_get_type ()

```
GType
ags_osc_xmlrpc_controller_get_type ();
```

Types and Values

AGS_OSC_XMLRPC_CONTROLLER_DEFAULT_DELEGATE_TIMEOUT

```
#define AGS_OSC_XMLRPC_CONTROLLER_DEFAULT_DELEGATE_TIMEOUT (1.0 / 30.0)
```

enum AgsOscXmlrpcControllerFlags

Enum values to control the behavior or indicate internal state of AgsOscXmlrpcController by enable/disable as flags.

Members

AGS_OSC_XMLRPC_CONTROLLER_DELEGATE_STARTED		
AGS_OSC_XMLRPC_CONTROLLER_DELEGATE_RUNNING		
AGS_OSC_XMLRPC_CONTROLLER_DELEGATE_TERMINATING		

AGS_TYPE_OSC_XMLRPC_CONTROLLER

```
#define AGS_TYPE_OSC_XMLRPC_CONTROLLER                (ags_osc_xmlrpc_controller_get_type() ↩
    )
```

struct AgsOscXmlrpcController

```
struct AgsOscXmlrpcController;
```

struct AgsOscXmlrpcControllerClass

```
struct AgsOscXmlrpcControllerClass {
  AgsControllerClass controller;

  void (*start_delegate)(AgsOscXmlrpcController *osc_xmlrpc_controller);
  void (*stop_delegate)(AgsOscXmlrpcController *osc_xmlrpc_controller);
};
```

Property Details

The "osc-xmlrpc-server" property

```
  "osc-xmlrpc-server"        AgsOscXmlrpcServer~*
```

The assigned AgsOscXmlrpcServer

Owner: AgsOscXmlrpcController

Flags: Read / Write

Since: 3.0.0

Signal Details

The "start-delegate" signal

```
void
user_function (AgsOscXmlrpcController *osc_xmlrpc_controller,
               gpointer                user_data)
```

The ::start-delegate signal is emited during start of delegating messages.

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop-delegate" signal

```
void
user_function (AgsOscXmlrpcController *osc_xmlrpc_controller,
               gpointer               user_data)
```

The ::stop-delegate signal is emited during stop of delegating messages.

Parameters

osc_xmlrpc_controller	the AgsOscXmlrpcController	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 18

Audio files - Reading/Writing audio data from disk

Perform audio file disk IO.

18.1 AgsAudioContainerManager

AgsAudioContainerManager — The audio container manager

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_CONTAINER_MANAGER_GET_OBJ_MUTEX()
GRecMutex *	ags_audio_container_manager_get_obj_mutex ()
void	ags_audio_container_manager_add_audio_container ()
void	ags_audio_container_manager_remove_audio_container ()
GObject *	ags_audio_container_manager_find_audio_container ()
AgsAudioContainerManager *	ags_audio_container_manager_get_instance ()
AgsAudioContainerManager *	ags_audio_container_manager_new ()
#define	AGS_AUDIO_CONTAINER_MANAGER()
#define	AGS_AUDIO_CONTAINER_MANAGER_CLASS()
#define	AGS_AUDIO_CONTAINER_MANAGER_GET_CLASS()
#define	AGS_IS_AUDIO_CONTAINER_MANAGER()
#define	AGS_IS_AUDIO_CONTAINER_MANAGER_CLASS()
GType	ags_audio_container_manager_get_type ()

Properties

gpointer	audio-container	Read / Write

Types and Values

#define	AGS_TYPE_AUDIO_CONTAINER_MANAGER

struct	AgsAudioContainerManager
struct	AgsAudioContainerManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsAudioContainerManager
```

Includes

```
#include <ags/audio_container/ags_audio_container_manager.h>
```

Description

AgsAudioContainerManager stores audio containers.

Functions

AGS_AUDIO_CONTAINER_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_CONTAINER_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsAudioContainerManager *) obj ↩
    )->obj_mutex))
```

ags_audio_container_manager_get_obj_mutex ()

```
GRecMutex~*
ags_audio_container_manager_get_obj_mutex
                                (AgsAudioContainerManager *audio_container_manager);
```

ags_audio_container_manager_add_audio_container ()

```
void
ags_audio_container_manager_add_audio_container
                                (AgsAudioContainerManager *audio_container_manager,
                                 GObject *audio_container);
```

Add *audio_container* to *audio_container_manager* .

Parameters

audio_container_manager	the AgsAudioContainer-Manager	
audio_container	the AgsAudioContainer	

Since: 3.4.0

ags_audio_container_manager_remove_audio_container ()

```
void
ags_audio_container_manager_remove_audio_container
                                (AgsAudioContainerManager *audio_container_manager,
                                 GObject *audio_container);
```

Remove *audio_container* from *audio_container_manager* .

Parameters

audio_container_manager	the AgsAudioContainer-Manager	
audio_container	the AgsAudioContainer	

Since: 3.4.0

ags_audio_container_manager_find_audio_container ()

```
GObject~*
ags_audio_container_manager_find_audio_container
                                (AgsAudioContainerManager *audio_container_manager,
                                 gchar *filename);
```

Find audio container by *filename* .

Parameters

audio_container_manager	the AgsAudioContainer-Manager	
filename	the string identifier	

Returns

the matching AgsAudioContainer if found, otherwise NULL.

[transfer full]

Since: 3.4.0

ags_audio_container_manager_get_instance ()

```
AgsAudioContainerManager~*
ags_audio_container_manager_get_instance
                                ();
```

Get ags audio container manager instance.

Returns

the AgsAudioContainerManager singleton.

[transfer none]

Since: 3.4.0

ags_audio_container_manager_new ()

```
AgsAudioContainerManager~*
ags_audio_container_manager_new ();
```

Creates an AgsAudioContainerManager

Returns

a new AgsAudioContainerManager

Since: 3.4.0

AGS_AUDIO_CONTAINER_MANAGER()

```
#define AGS_AUDIO_CONTAINER_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_AUDIO_CONTAINER_MANAGER, AgsAudioContainerManager))
```

AGS_AUDIO_CONTAINER_MANAGER_CLASS()

```
#define AGS_AUDIO_CONTAINER_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class, ↩
    AGS_TYPE_AUDIO_CONTAINER_MANAGER, AgsAudioContainerManagerClass))
```

AGS_AUDIO_CONTAINER_MANAGER_GET_CLASS()

```
#define AGS_AUDIO_CONTAINER_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj, ↩
    AGS_TYPE_AUDIO_CONTAINER_MANAGER, AgsAudioContainerManagerClass))
```

AGS_IS_AUDIO_CONTAINER_MANAGER()

```
#define AGS_IS_AUDIO_CONTAINER_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
     AGS_TYPE_AUDIO_CONTAINER_MANAGER))
```

AGS_IS_AUDIO_CONTAINER_MANAGER_CLASS()

```
#define AGS_IS_AUDIO_CONTAINER_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
    AGS_TYPE_AUDIO_CONTAINER_MANAGER))
```

ags_audio_container_manager_get_type ()

```
GType
ags_audio_container_manager_get_type ();
```

Types and Values

AGS_TYPE_AUDIO_CONTAINER_MANAGER

```
#define AGS_TYPE_AUDIO_CONTAINER_MANAGER                ( ↩
    ags_audio_container_manager_get_type())
```

struct AgsAudioContainerManager

```
struct AgsAudioContainerManager;
```

struct AgsAudioContainerManagerClass

```
struct AgsAudioContainerManagerClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-container" property

```
  "audio-container"          gpointer
```

The GList containing AgsAudioContainer.

Owner: AgsAudioContainerManager

Flags: Read / Write

Since: 3.4.0

18.2 AgsAudioContainer

AgsAudioContainer — Audio container input/output

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_CONTAINER_GET_OBJ_MUTEX()
gboolean	ags_audio_container_test_flags ()
void	ags_audio_container_set_flags ()
void	ags_audio_container_unset_flags ()
gboolean	ags_audio_container_check_suffix ()
GList *	ags_audio_container_find_sound_resource ()
void	ags_audio_container_add_audio_signal ()
void	ags_audio_container_remove_audio_signal ()
void	ags_audio_container_add_wave ()
void	ags_audio_container_remove_wave ()
gboolean	ags_audio_container_open ()
gboolean	ags_audio_container_open_from_data ()
gboolean	ags_audio_container_rw_open ()
void	ags_audio_container_close ()
void *	ags_audio_container_read ()
GList *	ags_audio_container_read_audio_signal ()
GList *	ags_audio_container_read_wave ()

void	ags_audio_container_seek ()
void	ags_audio_container_write ()
void	ags_audio_container_flush ()
AgsAudioContainer *	ags_audio_container_new ()
#define	AGS_AUDIO_CONTAINER()
#define	AGS_AUDIO_CONTAINER_CLASS()
#define	AGS_AUDIO_CONTAINER_GET_CLASS()
#define	AGS_IS_AUDIO_CONTAINER()
#define	AGS_IS_AUDIO_CONTAINER_CLASS()
GType	ags_audio_container_get_type ()

Properties

int	audio-channel	Read / Write
gpointer	audio-signal	Read / Write
guint	buffer-size	Read / Write
guint	file-audio-channels	Read / Write
guint	file-frame-count	Read / Write
guint	file-samplerate	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
char *	instrument	Read / Write
char *	preset	Read / Write
char *	sample	Read / Write
guint	samplerate	Read / Write
GObject *	soundcard	Read / Write
gpointer	wave	Read / Write

Types and Values

#define	AGS_TYPE_AUDIO_CONTAINER
struct	AgsAudioContainer
struct	AgsAudioContainerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsAudioContainer
```

Implemented Interfaces

AgsAudioContainer implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_audio_container.h>
```

Description

AgsAudioContainer is the base object to read/write audio data.

Functions

AGS_AUDIO_CONTAINER_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_CONTAINER_GET_OBJ_MUTEX(obj) (&(((AgsAudioContainer *) obj)->obj_mutex))
```

ags_audio_container_test_flags ()

```
gboolean
ags_audio_container_test_flags (AgsAudioContainer *audio_container,
                                guint flags);
```

Test *flags* to be set on *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_container_set_flags ()

```
void
ags_audio_container_set_flags (AgsAudioContainer *audio_container,
                               guint flags);
```

Enable a feature of *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
flags	see AgsAudioContainerFlags	

Since: 3.0.0

ags_audio_container_unset_flags ()

```
void
ags_audio_container_unset_flags (AgsAudioContainer *audio_container,
                                 guint flags);
```

Disable a feature of *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
flags	see AgsAudioContainerFlags	

Since: 3.0.0

ags_audio_container_check_suffix ()

```
gboolean
ags_audio_container_check_suffix (gchar *filename);
```

Check suffix.

Parameters

filename	the filename	

Returns

TRUE if suffix supported, else FALSE

Since: 3.0.0

ags_audio_container_find_sound_resource ()

```
GList~*
ags_audio_container_find_sound_resource
                                (AgsAudioContainer *audio_container,
                                 gchar *preset,
                                 gchar *instrument,
                                 gchar *sample);
```

Find resource specified by parameters.

Parameters

audio_container	the AgsAudioContainer	
preset	the preset	
instrument	the instrument	
sample	the sample	

Returns

the GList containing GObject sample or NULL. WARNING the parameters need to be valid in order to return the correct list.

[element-type GObject][transfer full]

Since: 3.0.0

ags_audio_container_add_audio_signal ()

```
void
ags_audio_container_add_audio_signal (AgsAudioContainer *audio_container,
                                      GObject *audio_signal);
```

Add *audio_signal* to *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
audio_signal	the AgsAudioSignal	

Since: 3.0.0

ags_audio_container_remove_audio_signal ()

```
void
ags_audio_container_remove_audio_signal
                                (AgsAudioContainer *audio_container,
                                 GObject *audio_signal);
```

Remove *audio_signal* from *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
audio_signal	the AgsAudioSignal	

Since: 3.0.0

ags_audio_container_add_wave ()

```
void
ags_audio_container_add_wave (AgsAudioContainer *audio_container,
                              GObject *wave);
```

Add *wave* to *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
wave	the AgsWave	

Since: 3.0.0

ags_audio_container_remove_wave ()

```
void
ags_audio_container_remove_wave (AgsAudioContainer *audio_container,
                                 GObject *wave);
```

Remove *wave* from *audio_container* .

Parameters

audio_container	the AgsAudioContainer	
wave	the AgsWave	

Since: 3.0.0

ags_audio_container_open ()

```
gboolean
ags_audio_container_open (AgsAudioContainer *audio_container);
```

Open the AgsAudioContainer in read mode.

Parameters

audio_container	the AgsAudioContainer	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_audio_container_open_from_data ()

```
gboolean
ags_audio_container_open_from_data (AgsAudioContainer *audio_container,
                                    gchar *data);
```

ags_audio_container_rw_open ()

```
gboolean
ags_audio_container_rw_open (AgsAudioContainer *audio_container,
                             gboolean create);
```

ags_audio_container_close ()

```
void
ags_audio_container_close (AgsAudioContainer *audio_container);
```

Close the AgsAudioContainer.

Parameters

audio_container	the AgsAudioContainer	

Since: 3.0.0

ags_audio_container_read ()

```
void~*
ags_audio_container_read (AgsAudioContainer *audio_container,
                          guint audio_channel,
                          guint format,
                          GError **error);
```

Read audio buffer.

Parameters

audio_container	the AgsAudioContainer	
audio_channel	nth channel	
format	the format	
error	returned error	

Since: 3.0.0

ags_audio_container_read_audio_signal ()

```
GList~*
ags_audio_container_read_audio_signal (AgsAudioContainer *audio_container);
```

Convert the AgsAudioContainer to a GList of AgsAudioSignal.

Parameters

audio_container	the AgsAudioContainer	

Returns

the new GList containing AgsAudioSignal or NULL.

[element-type AgsAudio.AudioSignal][transfer full]

Since: 3.0.0

ags_audio_container_read_wave ()

```
GList~*
ags_audio_container_read_wave (AgsAudioContainer *audio_container,
                               guint64 x_offset,
                               gdouble delay,
                               guint attack);
```

Convert the AgsAudioContainer to a GList of AgsWave.

Parameters

audio_container	the AgsAudioContainer	
x_offset	the x offset	
delay	the delay	
attack	the attack	

Returns

the new GList containing AgsWave or NULL.

[element-type AgsAudio.Wave][transfer full]

Since: 3.0.0

ags_audio_container_seek ()

```
void
ags_audio_container_seek (AgsAudioContainer *audio_container,
                          guint frames,
                          gint whence);
```

ags_audio_container_write ()

```
void
ags_audio_container_write (AgsAudioContainer *audio_container,
                           void *buffer,
                           guint buffer_size,
                           guint format);
```

ags_audio_container_flush ()

```
void
ags_audio_container_flush (AgsAudioContainer *audio_container);
```

ags_audio_container_new ()

```
AgsAudioContainer~*
ags_audio_container_new (gchar *filename,
                         gchar *preset,
                         gchar *instrument,
                         gchar *sample,
                         GObject *soundcard,
                         gint audio_channel);
```

Create a new instance of AgsAudioContainer.

Parameters

filename	the filename	
preset	the preset	
instrument	the instrument	
sample	the sample	
soundcard	defaults of AgsSoundcard	
audio_channel	the audio channel to read	

Returns

the new AgsAudioContainer.

Since: 3.0.0

AGS_AUDIO_CONTAINER()

```
#define AGS_AUDIO_CONTAINER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_CONTAINER, AgsAudioContainer))
```

AGS_AUDIO_CONTAINER_CLASS()

```
#define AGS_AUDIO_CONTAINER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_AUDIO_CONTAINER, AgsAudioContainerClass))
```

AGS_AUDIO_CONTAINER_GET_CLASS()

```
#define AGS_AUDIO_CONTAINER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_AUDIO_CONTAINER, AgsAudioContainerClass))
```

AGS_IS_AUDIO_CONTAINER()

```
#define AGS_IS_AUDIO_CONTAINER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_AUDIO_CONTAINER))
```

AGS_IS_AUDIO_CONTAINER_CLASS()

```
#define AGS_IS_AUDIO_CONTAINER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_AUDIO_CONTAINER))
```

ags_audio_container_get_type ()

```
GType
ags_audio_container_get_type ();
```

Types and Values

AGS_TYPE_AUDIO_CONTAINER

```
#define AGS_TYPE_AUDIO_CONTAINER                (ags_audio_container_get_type())
```

struct AgsAudioContainer

```
struct AgsAudioContainer;
```

struct AgsAudioContainerClass

```
struct AgsAudioContainerClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            int
```

The audio channel to be read.

Owner: AgsAudioContainer

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "audio-signal" property

```
  "audio-signal"             gpointer
```

The containing AgsAudioSignal.

[transfer full]

Owner: AgsAudioContainer

Flags: Read / Write

Since: 3.0.0

The "buffer-size" property

```
  "buffer-size"              guint
```

The buffer size to be used.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "file-audio-channels" property

```
"file-audio-channels"      guint
```

The audio channel count of this file.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "file-frame-count" property

```
"file-frame-count"         guint
```

The frame count of the file.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: 0

The "file-samplerate" property

```
"file-samplerate"          guint
```

The samplerate of this file.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "filename" property

```
"filename"                 char~*
```

The assigned filename.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "format" property

```
"format"                   guint
```

The format to be used.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "instrument" property

```
"instrument"                char~*
```

The assigned instrument.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "preset" property

```
"preset"                    char~*
```

The assigned preset.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "sample" property

```
"sample"                    char~*
```

The assigned sample.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsAudioContainer

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "soundcard" property

```
"soundcard"                 GObject~*
```

The assigned soundcard.

Owner: AgsAudioContainer

Flags: Read / Write

Since: 3.0.0

The "wave" property

```
"wave"                    gpointer
```

The containing AgsWave.

[transfer full]

Owner: AgsAudioContainer

Flags: Read / Write

Since: 3.0.0

18.3 AgsAudioFileManager

AgsAudioFileManager — The audio file manager

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_FILE_MANAGER_GET_OBJ_MUTEX()
GRecMutex *	ags_audio_file_manager_get_obj_mutex ()
void	ags_audio_file_manager_add_audio_file ()
void	ags_audio_file_manager_remove_audio_file ()
GObject *	ags_audio_file_manager_find_audio_file ()
AgsAudioFileManager *	ags_audio_file_manager_get_instance ()
AgsAudioFileManager *	ags_audio_file_manager_new ()
#define	AGS_AUDIO_FILE_MANAGER()
#define	AGS_AUDIO_FILE_MANAGER_CLASS()
#define	AGS_AUDIO_FILE_MANAGER_GET_CLASS()
#define	AGS_IS_AUDIO_FILE_MANAGER()
#define	AGS_IS_AUDIO_FILE_MANAGER_CLASS()
GType	ags_audio_file_manager_get_type ()

Properties

gpointer	audio-file	Read / Write

Types and Values

#define	AGS_TYPE_AUDIO_FILE_MANAGER
struct	AgsAudioFileManager
struct	AgsAudioFileManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsAudioFileManager
```

Includes

```
#include <ags/audio_file/ags_audio_file_manager.h>
```

Description

AgsAudioFileManager stores audio files.

Functions

AGS_AUDIO_FILE_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_FILE_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsAudioFileManager *) obj)-> ↩
    obj_mutex))
```

ags_audio_file_manager_get_obj_mutex ()

```
GRecMutex~*
ags_audio_file_manager_get_obj_mutex (AgsAudioFileManager *audio_file_manager);
```

ags_audio_file_manager_add_audio_file ()

```
void
ags_audio_file_manager_add_audio_file (AgsAudioFileManager *audio_file_manager,
                                       GObject *audio_file);
```

Add *audio_file* to *audio_file_manager* .

Parameters

audio_file_manager	the AgsAudioFileManager	
audio_file	the AgsAudioFile	

Since: 3.4.0

ags_audio_file_manager_remove_audio_file ()

```
void
ags_audio_file_manager_remove_audio_file
                                (AgsAudioFileManager *audio_file_manager,
                                 GObject *audio_file);
```

Remove *audio_file* from *audio_file_manager* .

Parameters

audio_file_manager	the AgsAudioFileManager	
audio_file	the AgsAudioFile	

Since: 3.4.0

ags_audio_file_manager_find_audio_file ()

```
GObject~*
ags_audio_file_manager_find_audio_file
                                (AgsAudioFileManager *audio_file_manager,
                                 gchar *filename);
```

Find audio file by *filename* .

Parameters

audio_file_manager	the AgsAudioFileManager	
filename	the string identifier	

Returns

the matching AgsAudioFile if found, otherwise NULL.

[transfer none]

Since: 3.4.0

ags_audio_file_manager_get_instance ()

```
AgsAudioFileManager~*
ags_audio_file_manager_get_instance ();
```

Get ags audio file manager instance.

Returns

the AgsAudioFileManager singleton.

[transfer none]

Since: 3.4.0

ags_audio_file_manager_new ()

```
AgsAudioFileManager~*
ags_audio_file_manager_new ();
```

Creates an AgsAudioFileManager

Returns

a new AgsAudioFileManager

Since: 3.4.0

AGS_AUDIO_FILE_MANAGER()

```
#define AGS_AUDIO_FILE_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_FILE_MANAGER, AgsAudioFileManager))
```

AGS_AUDIO_FILE_MANAGER_CLASS()

```
#define AGS_AUDIO_FILE_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_FILE_MANAGER, AgsAudioFileManagerClass))
```

AGS_AUDIO_FILE_MANAGER_GET_CLASS()

```
#define AGS_AUDIO_FILE_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_FILE_MANAGER, AgsAudioFileManagerClass))
```

AGS_IS_AUDIO_FILE_MANAGER()

```
#define AGS_IS_AUDIO_FILE_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_FILE_MANAGER))
```

AGS_IS_AUDIO_FILE_MANAGER_CLASS()

```
#define AGS_IS_AUDIO_FILE_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_FILE_MANAGER))
```

ags_audio_file_manager_get_type ()

```
GType
ags_audio_file_manager_get_type ();
```

Types and Values

AGS_TYPE_AUDIO_FILE_MANAGER

```
#define AGS_TYPE_AUDIO_FILE_MANAGER                (ags_audio_file_manager_get_type())
```

struct AgsAudioFileManager

```
struct AgsAudioFileManager;
```

struct AgsAudioFileManagerClass

```
struct AgsAudioFileManagerClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-file" property

```
"audio-file"                gpointer
```

The GList containing AgsAudioFile.

Owner: AgsAudioFileManager

Flags: Read / Write

Since: 3.4.0

18.4 AgsAudioFile

AgsAudioFile — Audio file input/output

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUDIO_FILE_GET_OBJ_MUTEX()
gboolean	ags_audio_file_test_flags ()
void	ags_audio_file_set_flags ()
void	ags_audio_file_unset_flags ()
gboolean	ags_audio_file_check_suffix ()
GObject *	ags_audio_file_get_soundcard ()
void	ags_audio_file_set_soundcard ()
gchar *	ags_audio_file_get_filename ()
void	ags_audio_file_set_filename ()
guint	ags_audio_file_get_file_audio_channels ()
void	ags_audio_file_set_file_audio_channels ()
guint	ags_audio_file_get_file_samplerate ()
void	ags_audio_file_set_file_samplerate ()
guint	ags_audio_file_get_file_frame_count ()
void	ags_audio_file_set_file_frame_count ()
guint	ags_audio_file_get_samplerate ()
void	ags_audio_file_set_samplerate ()
guint	ags_audio_file_get_frame_count ()
void	ags_audio_file_set_frame_count ()
guint	ags_audio_file_get_format ()
void	ags_audio_file_set_format ()
guint	ags_audio_file_get_audio_channel ()
void	ags_audio_file_set_audio_channel ()
GObject *	ags_audio_file_get_sound_resource ()
void	ags_audio_file_set_sound_resource ()
void	ags_audio_file_add_audio_signal ()
void	ags_audio_file_remove_audio_signal ()
void	ags_audio_file_add_wave ()
void	ags_audio_file_remove_wave ()
gboolean	ags_audio_file_open ()

gboolean	ags_audio_file_open_from_data ()
gboolean	ags_audio_file_rw_open ()
void	ags_audio_file_close ()
void *	ags_audio_file_read ()
void	ags_audio_file_read_audio_signal ()
void	ags_audio_file_read_wave ()
void	ags_audio_file_seek ()
void	ags_audio_file_write ()
void	ags_audio_file_flush ()
AgsAudioFile *	ags_audio_file_new ()
#define	AGS_AUDIO_FILE()
#define	AGS_AUDIO_FILE_CLASS()
#define	AGS_AUDIO_FILE_GET_CLASS()
#define	AGS_IS_AUDIO_FILE()
#define	AGS_IS_AUDIO_FILE_CLASS()
GType	ags_audio_file_get_type ()

Properties

int	audio-channel	Read / Write
gpointer	audio-signal	Read / Write
guint	buffer-size	Read / Write
guint	file-audio-channels	Read / Write
guint	file-frame-count	Read / Write
guint	file-samplerate	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
guint	samplerate	Read / Write
GObject *	soundcard	Read / Write
gpointer	wave	Read / Write

Types and Values

#define	AGS_TYPE_AUDIO_FILE
struct	AgsAudioFile
struct	AgsAudioFileClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsAudioFile
```

Implemented Interfaces

AgsAudioFile implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_audio_file.h>
```

Description

AgsAudioFile is the base object to read/write audio data.

Functions

AGS_AUDIO_FILE_GET_OBJ_MUTEX()

```
#define AGS_AUDIO_FILE_GET_OBJ_MUTEX(obj) (&(((AgsAudioFile *) obj)->obj_mutex))
```

ags_audio_file_test_flags ()

```
gboolean
ags_audio_file_test_flags (AgsAudioFile *audio_file,
                           guint flags);
```

Test *flags* to be set on *audio_file* .

Parameters

audio_file	the AgsAudioFile	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_audio_file_set_flags ()

```
void
ags_audio_file_set_flags (AgsAudioFile *audio_file,
                          guint flags);
```

Enable a feature of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
flags	see AgsAudioFileFlags	

Since: 3.0.0

ags_audio_file_unset_flags ()

```
void
ags_audio_file_unset_flags (AgsAudioFile *audio_file,
                            guint flags);
```

Disable a feature of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
flags	see AgsAudioFileFlags	

Since: 3.0.0

ags_audio_file_check_suffix ()

```
gboolean
ags_audio_file_check_suffix (gchar *filename);
```

Check suffix.

Parameters

filename	the filename	

Returns

TRUE if suffix supported, else FALSE

Since: 3.0.0

ags_audio_file_get_soundcard ()

```
GObject~*
ags_audio_file_get_soundcard (AgsAudioFile *audio_file);
```

Get soundcard of *audio_file* .

Parameters

audio_file	the AgsAudioFile	

Returns

the GObject implementing AgsSoundcard.

[transfer full]

Since: 4.2.0

ags_audio_file_set_soundcard ()

```
void
ags_audio_file_set_soundcard (AgsAudioFile *audio_file,
                              GObject *soundcard);
```

Set soundcard of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
soundcard	the GObject implementing AgsSoundcard	

Since: 4.2.0

ags_audio_file_get_filename ()

```
gchar~*
ags_audio_file_get_filename (AgsAudioFile *audio_file);
```

Get filename of *audio_file* .

Parameters

audio_file	the AgsAudioFile	

Returns

the filename

Since: 4.2.0

ags_audio_file_set_filename ()

```
void
ags_audio_file_set_filename (AgsAudioFile *audio_file,
                             gchar *filename);
```

Set filename of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
filename	the filename	

Since: 4.2.0

ags_audio_file_get_file_audio_channels ()

```
guint
ags_audio_file_get_file_audio_channels
                                (AgsAudioFile *audio_file);
```

Get file audio channels of *audio_file* .

Parameters

audio_file	the AgsAudioFile

Returns

the file audio channels

Since: 4.2.0

ags_audio_file_set_file_audio_channels ()

```
void
ags_audio_file_set_file_audio_channels
                                (AgsAudioFile *audio_file,
                                 guint file_audio_channels);
```

Get file audio channels of *audio_file* .

Parameters

audio_file	the AgsAudioFile
file_audio_channels	the file audio channels

Since: 4.2.0

ags_audio_file_get_file_samplerate ()

```
guint
ags_audio_file_get_file_samplerate (AgsAudioFile *audio_file);
```

Get file samplerate of *audio_file* .

Parameters

audio_file	the AgsAudioFile

Returns

the file samplerate

Since: 4.2.0

ags_audio_file_set_file_samplerate ()

```
void
ags_audio_file_set_file_samplerate (AgsAudioFile *audio_file,
                                    guint file_samplerate);
```

Get file samplerate of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
file_samplerate	the file samplerate	

Since: 4.2.0

ags_audio_file_get_file_frame_count ()

```
guint
ags_audio_file_get_file_frame_count (AgsAudioFile *audio_file);
```

Get file frame count of *audio_file* .

Parameters

audio_file	the AgsAudioFile	

Returns

the file frame count

Since: 4.2.0

ags_audio_file_set_file_frame_count ()

```
void
ags_audio_file_set_file_frame_count (AgsAudioFile *audio_file,
                                     guint file_frame_count);
```

Get file frame count of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
file_frame_count	the file frame count	

Since: 4.2.0

ags_audio_file_get_samplerate ()

```
guint
ags_audio_file_get_samplerate (AgsAudioFile *audio_file);
```

Get samplerate of *audio_file* .

Parameters

audio_file	the AgsAudioFile	

Returns

the samplerate

Since: 4.2.0

ags_audio_file_set_samplerate ()

```
void
ags_audio_file_set_samplerate (AgsAudioFile *audio_file,
                               guint samplerate);
```

Get samplerate of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
samplerate	the samplerate	

Since: 4.2.0

ags_audio_file_get_frame_count ()

```
guint
ags_audio_file_get_frame_count (AgsAudioFile *audio_file);
```

Get frame_count of *audio_file* .

Parameters

audio_file	the AgsAudioFile	

Returns

the frame_count

Since: 4.2.0

ags_audio_file_set_frame_count ()

```
void
ags_audio_file_set_frame_count (AgsAudioFile *audio_file,
                                guint frame_count);
```

Get frame_count of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
frame_count	the frame_count	

Since: 4.2.0

ags_audio_file_get_format ()

```
guint
ags_audio_file_get_format (AgsAudioFile *audio_file);
```

Get format of *audio_file* .

Parameters

audio_file	the AgsAudioFile

Returns

the format

Since: 4.2.0

ags_audio_file_set_format ()

```
void
ags_audio_file_set_format (AgsAudioFile *audio_file,
                           guint format);
```

Get format of *audio_file* .

Parameters

audio_file	the AgsAudioFile
format	the format

Since: 4.2.0

ags_audio_file_get_audio_channel ()

```
guint
ags_audio_file_get_audio_channel (AgsAudioFile *audio_file);
```

Get audio channel of *audio_file* .

Parameters

audio_file	the AgsAudioFile

Returns

the audio channel

Since: 4.2.0

ags_audio_file_set_audio_channel ()

```
void
ags_audio_file_set_audio_channel (AgsAudioFile *audio_file,
                                  gint audio_channel);
```

Get audio channel of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
audio_channel	the audio channel	

Since: 4.2.0

ags_audio_file_get_sound_resource ()

```
GObject~*
ags_audio_file_get_sound_resource (AgsAudioFile *audio_file);
```

Get sound resource of *audio_file* .

Parameters

audio_file	the AgsAudioFile	

Returns

the GObject implementing AgsSoundResource.

[transfer full]

Since: 4.2.0

ags_audio_file_set_sound_resource ()

```
void
ags_audio_file_set_sound_resource (AgsAudioFile *audio_file,
                                   GObject *sound_resource);
```

Set sound resource of *audio_file* .

Parameters

audio_file	the AgsAudioFile	
sound_resource	the GObject implementing AgsSoundResource	

Since: 4.2.0

ags_audio_file_add_audio_signal ()

```
void
ags_audio_file_add_audio_signal (AgsAudioFile *audio_file,
                                 GObject *audio_signal);
```

Add *audio_signal* to *audio_file*.

Parameters

audio_file	the AgsAudioFile	
audio_signal	the AgsAudioSignal	

Since: 3.0.0

ags_audio_file_remove_audio_signal ()

```
void
ags_audio_file_remove_audio_signal (AgsAudioFile *audio_file,
                                    GObject *audio_signal);
```

Remove *audio_signal* from *audio_file*.

Parameters

audio_file	the AgsAudioFile	
audio_signal	the AgsAudioSignal	

Since: 3.0.0

ags_audio_file_add_wave ()

```
void
ags_audio_file_add_wave (AgsAudioFile *audio_file,
                         GObject *wave);
```

Add *wave* to *audio_file*.

Parameters

audio_file	the AgsAudioFile	
wave	the AgsWave	

Since: 3.0.0

ags_audio_file_remove_wave ()

```
void
ags_audio_file_remove_wave (AgsAudioFile *audio_file,
                            GObject *wave);
```

Remove *wave* from *audio_file*.

Parameters

audio_file	the AgsAudioFile	
wave	the AgsWave	

Since: 3.0.0

ags_audio_file_open ()

```
gboolean
ags_audio_file_open (AgsAudioFile *audio_file);
```

Open the AgsAudioFile in read mode.

Parameters

audio_file	the AgsAudioFile	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_audio_file_open_from_data ()

```
gboolean
ags_audio_file_open_from_data (AgsAudioFile *audio_file,
                               gchar *data);
```

Open AgsAudioFile using virtual functions.

Parameters

audio_file	the AgsAudioFile	
data	the audio data	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_audio_file_rw_open ()

```
gboolean
ags_audio_file_rw_open (AgsAudioFile *audio_file,
                        gboolean create);
```

Open the AgsAudioFile in read/write mode.

Parameters

audio_file	the AgsAudioFile	
create	create the file	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_audio_file_close ()

```
void
ags_audio_file_close (AgsAudioFile *audio_file);
```

Close the AgsAudioFile.

Parameters

audio_file	the AgsAudioFile	

Since: 3.0.0

ags_audio_file_read ()

```
void~*
ags_audio_file_read (AgsAudioFile *audio_file,
                     guint audio_channel,
                     guint format,
                     GError **error);
```

Read audio buffer.

Parameters

audio_file	the AgsAudioFile	
audio_channel	nth channel	
format	the format	
error	returned error	

Since: 3.0.0

ags_audio_file_read_audio_signal ()

```
void
ags_audio_file_read_audio_signal (AgsAudioFile *audio_file);
```

Convert the AgsAudioFile to a GList of AgsAudioSignal.

Parameters

audio_file	the AgsAudioFile	

Since: 3.0.0

ags_audio_file_read_wave ()

```
void
ags_audio_file_read_wave (AgsAudioFile *audio_file,
                          guint64 x_offset,
                          gdouble delay,
                          guint attack);
```

Convert the AgsAudioFile to a GList of buffers.

Parameters

audio_file	the AgsAudioFile	
x_offset	the x offset	
delay	the delay	
attack	the attack	

Since: 3.0.0

ags_audio_file_seek ()

```
void
ags_audio_file_seek (AgsAudioFile *audio_file,
                     guint frames,
                     gint whence);
```

Position the AgsAudioFile's internal data address.

Parameters

audio_file	the AgsAudioFile	
frames	number of frames to seek	
whence	SEEK_SET, SEEK_CUR, or SEEK_END	

Since: 3.0.0

ags_audio_file_write ()

```
void
ags_audio_file_write (AgsAudioFile *audio_file,
                      void *buffer,
                      guint buffer_size,
                      guint format);
```

Write the buffer to AgsAudioFile.

Parameters

audio_file	the AgsAudioFile	
buffer	the audio data	
buffer_size	the count of frames to write	
format	the format	

Since: 3.0.0

ags_audio_file_flush ()

```
void
ags_audio_file_flush (AgsAudioFile *audio_file);
```

Flushes the AgsAudioFile's internal buffer.

Parameters

audio_file	the AgsAudioFile	

Since: 3.0.0

ags_audio_file_new ()

```
AgsAudioFile~*
ags_audio_file_new (gchar *filename,
                    GObject *soundcard,
                    gint audio_channel);
```

Create a new instance of AgsAudioFile.

Parameters

filename	the filename	
soundcard	defaults of AgsSoundcard	
audio_channel	the audio channel to read	

Returns

the new AgsAudioFile.

Since: 3.0.0

AGS_AUDIO_FILE()

```
#define AGS_AUDIO_FILE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_FILE, AgsAudioFile))
```

AGS_AUDIO_FILE_CLASS()

```
#define AGS_AUDIO_FILE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_AUDIO_FILE, AgsAudioFileClass))
```

AGS_AUDIO_FILE_GET_CLASS()

```
#define AGS_AUDIO_FILE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_AUDIO_FILE, AgsAudioFileClass))
```

AGS_IS_AUDIO_FILE()

```
#define AGS_IS_AUDIO_FILE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_AUDIO_FILE))
```

AGS_IS_AUDIO_FILE_CLASS()

```
#define AGS_IS_AUDIO_FILE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_AUDIO_FILE))
```

ags_audio_file_get_type ()

```
GType
ags_audio_file_get_type ();
```

Types and Values

AGS_TYPE_AUDIO_FILE

```
#define AGS_TYPE_AUDIO_FILE                 (ags_audio_file_get_type())
```

struct AgsAudioFile

```
struct AgsAudioFile;
```

struct AgsAudioFileClass

```
struct AgsAudioFileClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-channel" property

```
"audio-channel"            int
```

The audio channel to be read.

Owner: AgsAudioFile

Flags: Read / Write

Allowed values: >= -1

Default value: 0

Since: 3.0.0

The "audio-signal" property

```
"audio-signal"             gpointer
```

The containing AgsAudioSignal.

[transfer full]

Owner: AgsAudioFile

Flags: Read / Write

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"              guint
```

The buffer size to be used.

Owner: AgsAudioFile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "file-audio-channels" property

```
"file-audio-channels"      guint
```

The audio channel count of this file.

Owner: AgsAudioFile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "file-frame-count" property

```
"file-frame-count"          guint
```

The frame count of the file.

Owner: AgsAudioFile

Flags: Read / Write

Default value: 0

The "file-samplerate" property

```
"file-samplerate"           guint
```

The samplerate of this file.

Owner: AgsAudioFile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsAudioFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The format to be used.

Owner: AgsAudioFile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsAudioFile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "soundcard" property

```
"soundcard"                GObject~*
```

The assigned soundcard.

Owner: AgsAudioFile

Flags: Read / Write

Since: 3.0.0

The "wave" property

```
"wave"                     gpointer
```

The containing AgsWave.

[transfer full]

Owner: AgsAudioFile

Flags: Read / Write

Since: 3.0.0

18.5 AgsAudioFileLink

AgsAudioFileLink — link objects read of file.

Stability Level

Stable, unless otherwise indicated

Functions

AgsAudioFileLink *	ags_audio_file_link_new ()
#define	AGS_AUDIO_FILE_LINK()
#define	AGS_AUDIO_FILE_LINK_CLASS()
#define	AGS_AUDIO_FILE_LINK_GET_CLASS()
#define	AGS_IS_AUDIO_FILE_LINK()
#define	AGS_IS_AUDIO_FILE_LINK_CLASS()
GType	ags_audio_file_link_get_type ()

Properties

guint	audio-channel	Read / Write
char *	instrument	Read / Write
char *	preset	Read / Write
char *	sample	Read / Write
GObject *	timestamp	Read / Write

Types and Values

#define	AGS_TYPE_AUDIO_FILE_LINK
struct	AgsAudioFileLink
struct	AgsAudioFileLinkClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFileLink
        &#x2570;&#x2500;&#x2500; AgsAudioFileLink
```

Implemented Interfaces

AgsAudioFileLink implements AgsPlugin.

Includes

```
#include <ags/file/ags_audio_file_link.h>
```

Description

The AgsAudioFileLink links read objects of file.

Functions

ags_audio_file_link_new ()

```
AgsAudioFileLink~*
ags_audio_file_link_new ();
```

Create a new instance of AgsAudioFileLink

Returns

the new AgsAudioFileLink

Since: 3.0.0

AGS_AUDIO_FILE_LINK()

```
#define AGS_AUDIO_FILE_LINK(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUDIO_FILE_LINK, AgsAudioFileLink))
```

AGS_AUDIO_FILE_LINK_CLASS()

```
#define AGS_AUDIO_FILE_LINK_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUDIO_FILE_LINK, AgsAudioFileLink))
```

AGS_AUDIO_FILE_LINK_GET_CLASS()

```
#define AGS_AUDIO_FILE_LINK_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUDIO_FILE_LINK, AgsAudioFileLinkClass))
```

AGS_IS_AUDIO_FILE_LINK()

```
#define AGS_IS_AUDIO_FILE_LINK(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUDIO_FILE_LINK))
```

AGS_IS_AUDIO_FILE_LINK_CLASS()

```
#define AGS_IS_AUDIO_FILE_LINK_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUDIO_FILE_LINK))
```

ags_audio_file_link_get_type ()

```
GType
ags_audio_file_link_get_type ();
```

Types and Values

AGS_TYPE_AUDIO_FILE_LINK

```
#define AGS_TYPE_AUDIO_FILE_LINK                (ags_audio_file_link_get_type())
```

struct AgsAudioFileLink

```
struct AgsAudioFileLink;
```

struct AgsAudioFileLinkClass

```
struct AgsAudioFileLinkClass {
  AgsFileLinkClass file_link;
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            guint
```

The assigned audio channel.

Owner: AgsAudioFileLink

Flags: Read / Write

Allowed values: <= 256

Default value: 0

Since: 3.0.0

The "instrument" property

```
"instrument"                char~*
```

The assigned instrument.

Owner: AgsAudioFileLink

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "preset" property

```
"preset"                    char~*
```

The assigned preset.

Owner: AgsAudioFileLink

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "sample" property

```
"sample"                    char~*
```

The assigned sample.

Owner: AgsAudioFileLink

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "timestamp" property

```
"timestamp"                 GObject~*
```

The assigned timestamp.

Owner: AgsAudioFileLink

Flags: Read / Write

Since: 3.0.0

18.6 AgsGstreamerFile

AgsGstreamerFile — gstreamer file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_GSTREAMER_FILE_GET_OBJ_MUTEX()
#define	AGS_GSTREAMER_FILE_CHUNK_FRAME_COUNT()
gboolean	ags_gstreamer_file_test_flags ()
void	ags_gstreamer_file_set_flags ()
void	ags_gstreamer_file_unset_flags ()
gboolean	ags_gstreamer_file_check_suffix ()
GstEncodingProfile *	ags_gstreamer_file_create_wav_encoding_profile ()
GstEncodingProfile *	ags_gstreamer_file_create_mp3_encoding_profile ()
GstEncodingProfile *	ags_gstreamer_file_create_aac_encoding_profile ()
GstEncodingProfile *	ags_gstreamer_file_create_mp4_encoding_profile ()
GstEncodingProfile *	ags_gstreamer_file_create_mkv_encoding_profile ()
GstEncodingProfile *	ags_gstreamer_file_create_webm_encoding_profile ()
GstEncodingProfile *	ags_gstreamer_file_create_mpeg_encoding_profile ()
gboolean	ags_gstreamer_file_detect_encoding_profile ()
AgsGstreamerFile *	ags_gstreamer_file_new ()
#define	AGS_GSTREAMER_FILE()
#define	AGS_GSTREAMER_FILE_CLASS()
#define	AGS_GSTREAMER_FILE_GET_CLASS()
#define	AGS_IS_GSTREAMER_FILE()
#define	AGS_IS_GSTREAMER_FILE_CLASS()
GType	ags_gstreamer_file_get_type ()

Properties

guint	audio-channels	Read / Write
guint	buffer-size	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
guint	samplerate	Read / Write

Types and Values

#define	AGS_GSTREAMER_FILE_DEFAULT_AUDIO_CHANNELS
#define	AGS_GSTREAMER_FILE_DEFAULT_FORMAT
#define	AGS_GSTREAMER_FILE_DEFAULT_FLUSH_IDLE
#define	AGS_GSTREAMER_FILE_DEFAULT_DISCOVERER_TIMEOUT
#define	AGS_TYPE_GSTREAMER_FILE
struct	AgsGstreamerFile
struct	AgsGstreamerFileClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsGstreamerFile
```

Implemented Interfaces

AgsGstreamerFile implements AgsConnectable and AgsSoundResource.

Includes

```
#include <ags/audio/file/ags_gstreamer_file.h>
```

Description

AgsGstreamerFile is the base object to ineract with libgstreamer-1.0 file support.

Functions

AGS_GSTREAMER_FILE_GET_OBJ_MUTEX()

```
#define AGS_GSTREAMER_FILE_GET_OBJ_MUTEX(obj) (&(((AgsGstreamerFile *) obj)->obj_mutex))
```

AGS_GSTREAMER_FILE_CHUNK_FRAME_COUNT()

```
#define AGS_GSTREAMER_FILE_CHUNK_FRAME_COUNT(cc) (cc * 256)
```

ags_gstreamer_file_test_flags ()

```
gboolean
ags_gstreamer_file_test_flags (AgsGstreamerFile *gstreamer_file,
                               guint flags);
```

Test *flags* to be set on *gstreamer_file* .

Parameters

gstreamer_file	the AgsGstreamerFile	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.6.0

ags_gstreamer_file_set_flags ()

```
void
ags_gstreamer_file_set_flags (AgsGstreamerFile *gstreamer_file,
                              guint flags);
```

Enable a feature of *gstreamer_file* .

Parameters

gstreamer_file	the AgsGstreamerFile	
flags	see AgsGstreamerFileFlags	

Since: 3.6.0

ags_gstreamer_file_unset_flags ()

```
void
ags_gstreamer_file_unset_flags (AgsGstreamerFile *gstreamer_file,
                                guint flags);
```

Disable a feature of *gstreamer_file*.

Parameters

gstreamer_file	the AgsGstreamerFile	
flags	see AgsGstreamerFileFlags	

Since: 3.6.0

ags_gstreamer_file_check_suffix ()

```
gboolean
ags_gstreamer_file_check_suffix (gchar *filename);
```

Check suffix.

Parameters

filename	the filename	

Returns

TRUE if suffix supported, else FALSE

Since: 3.6.0

ags_gstreamer_file_create_wav_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_wav_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create wav rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile	

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_create_mp3_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_mp3_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create mp3 rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_create_aac_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_aac_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create aac rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_create_mp4_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_mp4_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create mp4 rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_create_mkv_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_mkv_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create mkv rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_create_webm_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_webm_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create webm rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_create_mpeg_encoding_profile ()

```
GstEncodingProfile~*
ags_gstreamer_file_create_mpeg_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Create mpeg rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Returns

the new GstEncodingProfile

Since: 3.6.0

ags_gstreamer_file_detect_encoding_profile ()

```
gboolean
ags_gstreamer_file_detect_encoding_profile
                                (AgsGstreamerFile *gstreamer_file);
```

Detect rw and create rw pipeline.

Parameters

gstreamer_file	the AgsGstreamerFile

Since: 3.6.0

ags_gstreamer_file_new ()

```
AgsGstreamerFile~*
ags_gstreamer_file_new ();
```

Creates a new instance of AgsGstreamerFile.

Returns

the new AgsGstreamerFile.

Since: 3.6.0

AGS_GSTREAMER_FILE()

```
#define AGS_GSTREAMER_FILE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GSTREAMER_FILE, AgsGstreamerFile))
```

AGS_GSTREAMER_FILE_CLASS()

```
#define AGS_GSTREAMER_FILE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_GSTREAMER_FILE, AgsGstreamerFileClass))
```

AGS_GSTREAMER_FILE_GET_CLASS()

```
#define AGS_GSTREAMER_FILE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_GSTREAMER_FILE, AgsGstreamerFileClass))
```

AGS_IS_GSTREAMER_FILE()

```
#define AGS_IS_GSTREAMER_FILE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_GSTREAMER_FILE))
```

AGS_IS_GSTREAMER_FILE_CLASS()

```
#define AGS_IS_GSTREAMER_FILE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_GSTREAMER_FILE))
```

ags_gstreamer_file_get_type ()

```
GType
ags_gstreamer_file_get_type ();
```

Types and Values

AGS_GSTREAMER_FILE_DEFAULT_AUDIO_CHANNELS

```
#define AGS_GSTREAMER_FILE_DEFAULT_AUDIO_CHANNELS (1)
```

AGS_GSTREAMER_FILE_DEFAULT_FORMAT

```
#define AGS_GSTREAMER_FILE_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_GSTREAMER_FILE_DEFAULT_FLUSH_IDLE

```
#define AGS_GSTREAMER_FILE_DEFAULT_FLUSH_IDLE (4)
```

AGS_GSTREAMER_FILE_DEFAULT_DISCOVERER_TIMEOUT

```
#define AGS_GSTREAMER_FILE_DEFAULT_DISCOVERER_TIMEOUT (3 * GST_SECOND)
```

AGS_TYPE_GSTREAMER_FILE

```
#define AGS_TYPE_GSTREAMER_FILE                 (ags_gstreamer_file_get_type())
```

struct AgsGstreamerFile

```
struct AgsGstreamerFile;
```

struct AgsGstreamerFileClass

```
struct AgsGstreamerFileClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-channels" property

```
"audio-channels"            guint
```

The audio channels to be used.

Owner: AgsGstreamerFile

Flags: Read / Write

Default value: 0

Since: 3.6.0

The "buffer-size" property

```
"buffer-size"               guint
```

The buffer size to be used.

Owner: AgsGstreamerFile

Flags: Read / Write

Default value: 0

Since: 3.6.0

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsGstreamerFile

Flags: Read / Write

Default value: NULL

Since: 3.6.0

The "format" property

```
"format"                    guint
```

The format to be used.

Owner: AgsGstreamerFile

Flags: Read / Write

Default value: 0

Since: 3.6.0

The "samplerate" property

```
"samplerate"                guint
```

The samplerate to be used.

Owner: AgsGstreamerFile

Flags: Read / Write

Default value: 0

Since: 3.6.0

18.7 AgsIpatch

AgsIpatch — Libinstpatch wrapper

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_IPATCH_GET_OBJ_MUTEX()
gboolean	ags_ipatch_test_flags ()
void	ags_ipatch_set_flags ()
void	ags_ipatch_unset_flags ()
gboolean	ags_ipatch_check_suffix ()
AgsIpatch *	ags_ipatch_new ()
#define	AGS_IPATCH()
#define	AGS_IPATCH_CLASS()
#define	AGS_IPATCH_GET_CLASS()
#define	AGS_IS_IPATCH()
#define	AGS_IS_IPATCH_CLASS()
GType	ags_ipatch_flags_get_type ()
GType	ags_ipatch_get_type ()

Properties

char *	filename	Read / Write
char *	mode	Read / Write
GObject *	soundcard	Read / Write

Types and Values

#define	AGS_IPATCH_DEFAULT_CHANNELS
#define	AGS_IPATCH_READ
#define	AGS_IPATCH_WRITE
enum	AgsIpatchFlags
#define	AGS_TYPE_IPATCH
struct	AgsIpatch

struct	AgsIpatchClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsIpatchFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsIpatch
```

Implemented Interfaces

AgsIpatch implements AgsConnectable and AgsSoundContainer.

Includes

```
#include <ags/audio/file/ags_ipatch.h>
```

Description

AgsIpatch is the base object to ineract with libinstpatch.

Functions

AGS_IPATCH_GET_OBJ_MUTEX()

```
#define AGS_IPATCH_GET_OBJ_MUTEX(obj) (&(((AgsIpatch *) obj)->obj_mutex))
```

ags_ipatch_test_flags ()

```
gboolean
ags_ipatch_test_flags (AgsIpatch *ipatch,
                       guint flags);
```

Test *flags* to be set on *ipatch* .

Parameters

ipatch	the AgsIpatch	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_ipatch_set_flags ()

```
void
ags_ipatch_set_flags (AgsIpatch *ipatch,
                      guint flags);
```

Enable a feature of *ipatch* .

Parameters

ipatch	the AgsIpatch	
flags	see AgsIpatchFlags	

Since: 3.0.0

ags_ipatch_unset_flags ()

```
void
ags_ipatch_unset_flags (AgsIpatch *ipatch,
                        guint flags);
```

Disable a feature of *ipatch* .

Parameters

ipatch	the AgsIpatch	
flags	see AgsIpatchFlags	

Since: 3.0.0

ags_ipatch_check_suffix ()

```
gboolean
ags_ipatch_check_suffix (gchar *filename);
```

Check *filename* 's suffix to be supported.

Parameters

filename	the filename	

Returns

TRUE if supported, else FALSE

Since: 3.0.0

ags_ipatch_new ()

```
AgsIpatch~*
ags_ipatch_new ();
```

Creates an AgsIpatch.

Returns

an empty AgsIpatch.

Since: 3.0.0

AGS_IPATCH()

```
#define AGS_IPATCH(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_IPATCH,  ↩
    AgsIpatch))
```

AGS_IPATCH_CLASS()

```
#define AGS_IPATCH_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_IPATCH,  ↩
    AgsIpatchClass))
```

AGS_IPATCH_GET_CLASS()

```
#define AGS_IPATCH_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_IPATCH,  ↩
    AgsIpatchClass))
```

AGS_IS_IPATCH()

```
#define AGS_IS_IPATCH(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_IPATCH))
```

AGS_IS_IPATCH_CLASS()

```
#define AGS_IS_IPATCH_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_IPATCH))
```

ags_ipatch_flags_get_type ()

```
GType
ags_ipatch_flags_get_type ();
```

ags_ipatch_get_type ()

```
GType
ags_ipatch_get_type ();
```

Types and Values

AGS_IPATCH_DEFAULT_CHANNELS

```
#define AGS_IPATCH_DEFAULT_CHANNELS (2)
```

AGS_IPATCH_READ

```
#define AGS_IPATCH_READ "r"
```

AGS_IPATCH_WRITE

```
#define AGS_IPATCH_WRITE "w"
```

enum AgsIpatchFlags

Enum values to control the behavior or indicate internal state of AgsFifoout by enable/disable as flags.

Members

AGS_IPATCH_DLS2	DLS2 format
AGS_IPATCH_SF2	Soundfont2 format
AGS_IPATCH_GIG	Gigasampler format

AGS_TYPE_IPATCH

```
#define AGS_TYPE_IPATCH                (ags_ipatch_get_type())
```

struct AgsIpatch

```
struct AgsIpatch;
```

struct AgsIpatchClass

```
struct AgsIpatchClass {
  GObjectClass gobject;
};
```

Property Details

The "filename" property

```
  "filename"                 char~*
```

The assigned filename.

Owner: AgsIpatch

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "mode" property

```
"mode"                    char~*
```

The assigned mode.

Owner: AgsIpatch

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "soundcard" property

```
"soundcard"               GObject~*
```

The assigned soundcard.

Owner: AgsIpatch

Flags: Read / Write

Since: 3.0.0

18.8 AgsIpatchGigReader

AgsIpatchGigReader — interfacing GIG related API of libinstpatch

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_IPATCH_GIG_READER_GET_OBJ_MUTEX()
gboolean	ags_ipatch_gig_reader_test_flags ()
void	ags_ipatch_gig_reader_set_flags ()
void	ags_ipatch_gig_reader_unset_flags ()
gboolean	ags_ipatch_gig_reader_load ()
gboolean	ags_ipatch_gig_reader_select_instrument ()
gboolean	ags_ipatch_gig_reader_select_sample ()
gchar **	ags_ipatch_gig_reader_get_instrument_all ()
gchar **	ags_ipatch_gig_reader_get_sample_all ()
gchar **	ags_ipatch_gig_reader_get_sample_by_instrument_index ()
AgsIpatchGigReader *	ags_ipatch_gig_reader_new ()
#define	AGS_IPATCH_GIG_READER()
#define	AGS_IPATCH_GIG_READER_CLASS()
#define	AGS_IPATCH_GIG_READER_GET_CLASS()
#define	AGS_IS_IPATCH_GIG_READER()
#define	AGS_IS_IPATCH_GIG_READER_CLASS()
GType	ags_ipatch_gig_reader_get_type ()

Properties

AgsIpatch *	ipatch	Read / Write

Types and Values

enum	AgsGigLevel
#define	AGS_TYPE_IPATCH_GIG_READER
struct	AgsIpatchGigReader
struct	AgsIpatchGigReaderClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsIpatchGigReader
```

Implemented Interfaces

AgsIpatchGigReader implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_ipatch_gig_reader.h>
```

Description

AgsIpatchGigReader is the base object to ineract with GIG related API.

Functions

AGS_IPATCH_GIG_READER_GET_OBJ_MUTEX()

```
#define AGS_IPATCH_GIG_READER_GET_OBJ_MUTEX(obj) (&(((AgsIpatchGigReader *) obj)->obj_mutex ↩
    ))
```

ags_ipatch_gig_reader_test_flags ()

```
gboolean
ags_ipatch_gig_reader_test_flags (AgsIpatchGigReader *ipatch_gig_reader,
                                  guint flags);
```

Test *flags* to be set on *ipatch_gig_reader* .

Parameters

ipatch_gig_reader	the AgsIpatchGigReader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_ipatch_gig_reader_set_flags ()

```
void
ags_ipatch_gig_reader_set_flags (AgsIpatchGigReader *ipatch_gig_reader,
                                 guint flags);
```

Enable a feature of *ipatch_gig_reader* .

Parameters

ipatch_gig_reader	the AgsIpatchGigReader	
flags	see AgsIpatchGigReaderFlags	

Since: 3.0.0

ags_ipatch_gig_reader_unset_flags ()

```
void
ags_ipatch_gig_reader_unset_flags (AgsIpatchGigReader *ipatch_gig_reader,
                                   guint flags);
```

Disable a feature of *ipatch_gig_reader* .

Parameters

ipatch_gig_reader	the AgsIpatchGigReader	
flags	see AgsIpatchGigReaderFlags	

Since: 3.0.0

ags_ipatch_gig_reader_load ()

```
gboolean
ags_ipatch_gig_reader_load (AgsIpatchGigReader *ipatch_gig_reader,
                            IpatchFileHandle *handle);
```

Load GIG file.

Parameters

ipatch_gig_reader	the AgsIpatchGigReader	
handle	the IpatchFileHandle	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_gig_reader_select_instrument ()

```
gboolean
ags_ipatch_gig_reader_select_instrument
                                (AgsIpatchGigReader *ipatch_gig_reader,
                                 guint instrument_index);
```

Select instrument.

Parameters

ipatch_gig_reader	the AgsIpatchGigReader	
instrument_index	the instrument index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_gig_reader_select_sample ()

```
gboolean
ags_ipatch_gig_reader_select_sample (AgsIpatchGigReader *ipatch_gig_reader,
                                     guint sample_index);
```

ags_ipatch_gig_reader_get_instrument_all ()

```
gchar~**
ags_ipatch_gig_reader_get_instrument_all
                                (AgsIpatchGigReader *ipatch_gig_reader);
```

Get instrument all.

Parameters

ipatch_gig_reader	the AgsIpatchGigReader	

Returns

the string vector containing instrument.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_gig_reader_get_sample_all ()

```
gchar~**
ags_ipatch_gig_reader_get_sample_all (AgsIpatchGigReader *ipatch_gig_reader);
```

Get sample all.

Parameters

ipatch_gig_reader	the AgsIpatchGigReader

Returns

the string vector containing sample.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_gig_reader_get_sample_by_instrument_index ()

```
gchar~**
ags_ipatch_gig_reader_get_sample_by_instrument_index
                                (AgsIpatchGigReader *ipatch_gig_reader,
                                 guint instrument_index);
```

Get sample by instrument index.

Parameters

ipatch_gig_reader	the AgsIpatchGigReader
instrument_index	the instrument index

Returns

the string vector containing sample.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_gig_reader_new ()

```
AgsIpatchGigReader~*
ags_ipatch_gig_reader_new (AgsIpatch *ipatch);
```

Creates a new instance of AgsIpatchGigReader.

Parameters

ipatch	the AgsIpatch

Returns

the new AgsIpatchGigReader.

Since: 3.0.0

AGS_IPATCH_GIG_READER()

```
#define AGS_IPATCH_GIG_READER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_IPATCH_GIG_READER, AgsIpatchGigReader))
```

AGS_IPATCH_GIG_READER_CLASS()

```
#define AGS_IPATCH_GIG_READER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_IPATCH_GIG_READER, AgsIpatchGigReaderClass))
```

AGS_IPATCH_GIG_READER_GET_CLASS()

```
#define AGS_IPATCH_GIG_READER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_IPATCH_GIG_READER, AgsIpatchGigReaderClass))
```

AGS_IS_IPATCH_GIG_READER()

```
#define AGS_IS_IPATCH_GIG_READER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_IPATCH_GIG_READER))
```

AGS_IS_IPATCH_GIG_READER_CLASS()

```
#define AGS_IS_IPATCH_GIG_READER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_IPATCH_GIG_READER))
```

ags_ipatch_gig_reader_get_type ()

```
GType
ags_ipatch_gig_reader_get_type ();
```

Types and Values

enum AgsGigLevel

Enum values to describe the different levels of a Soundfont2 file.

Members

AGS_GIG_FILENAME	filename
AGS_GIG_IHDR	instrument header
AGS_GIG_SHDR	sample header

AGS_TYPE_IPATCH_GIG_READER

```
#define AGS_TYPE_IPATCH_GIG_READER                (ags_ipatch_gig_reader_get_type())
```

struct AgsIpatchGigReader

```
struct AgsIpatchGigReader;
```

struct AgsIpatchGigReaderClass

```
struct AgsIpatchGigReaderClass {
  GObjectClass gobject;
};
```

Property Details

The "ipatch" property

```
  "ipatch"                    AgsIpatch~*
```

The assigned AgsIpatch

Owner: AgsIpatchGigReader

Flags: Read / Write

Since: 3.0.0

18.9 AgsIpatchSF2Reader

AgsIpatchSF2Reader — interfacing Soundfont2 related API of libinstpatch

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_IPATCH_SF2_READER_GET_OBJ_MUTEX()
gboolean	ags_ipatch_sf2_reader_test_flags ()
void	ags_ipatch_sf2_reader_set_flags ()
void	ags_ipatch_sf2_reader_unset_flags ()
gboolean	ags_ipatch_sf2_reader_load ()
gboolean	ags_ipatch_sf2_reader_select_preset ()
gboolean	ags_ipatch_sf2_reader_select_instrument ()
gboolean	ags_ipatch_sf2_reader_select_sample ()
gchar **	ags_ipatch_sf2_reader_get_preset_all ()
gchar **	ags_ipatch_sf2_reader_get_instrument_all ()
gchar **	ags_ipatch_sf2_reader_get_sample_all ()
gchar **	ags_ipatch_sf2_reader_get_instrument_by_preset_index ()
gchar **	ags_ipatch_sf2_reader_get_sample_by_preset_index ()
gchar **	ags_ipatch_sf2_reader_get_sample_by_preset_and_instrument_index ()
AgsIpatchSF2Reader *	ags_ipatch_sf2_reader_new ()
#define	AGS_IPATCH_SF2_READER()
#define	AGS_IPATCH_SF2_READER_CLASS()
#define	AGS_IPATCH_SF2_READER_GET_CLASS()
#define	AGS_IS_IPATCH_SF2_READER()
#define	AGS_IS_IPATCH_SF2_READER_CLASS()
GType	ags_ipatch_sf2_reader_get_type ()

Properties

AgsIpatch *	ipatch	Read / Write

Types and Values

enum	AgsSF2Level
#define	AGS_TYPE_IPATCH_SF2_READER
struct	AgsIpatchSF2Reader
struct	AgsIpatchSF2ReaderClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsIpatchSF2Reader
```

Implemented Interfaces

AgsIpatchSF2Reader implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_ipatch_sf2_reader.h>
```

Description

AgsIpatchSF2Reader is the base object to ineract with Soundfont2 related API.

Functions

AGS_IPATCH_SF2_READER_GET_OBJ_MUTEX()

```
#define AGS_IPATCH_SF2_READER_GET_OBJ_MUTEX(obj) (&(((AgsIpatchSF2Reader *) obj)->obj_mutex ↩
    ))
```

ags_ipatch_sf2_reader_test_flags ()

```
gboolean
ags_ipatch_sf2_reader_test_flags (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                  guint flags);
```

Test *flags* to be set on *ipatch_sf2_reader* .

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_ipatch_sf2_reader_set_flags ()

```
void
ags_ipatch_sf2_reader_set_flags (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                 guint flags);
```

Enable a feature of *ipatch_sf2_reader* .

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
flags	see AgsIpatchSF2ReaderFlags	

Since: 3.0.0

ags_ipatch_sf2_reader_unset_flags ()

```
void
ags_ipatch_sf2_reader_unset_flags (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                   guint flags);
```

Disable a feature of *ipatch_sf2_reader* .

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
flags	see AgsIpatchSF2ReaderFlags	

Since: 3.0.0

ags_ipatch_sf2_reader_load ()

```
gboolean
ags_ipatch_sf2_reader_load (AgsIpatchSF2Reader *ipatch_sf2_reader,
                            IpatchFileHandle *handle);
```

Load Soundfont2 file.

Parameters

ipatch_sf2_reader	the AgsSF2Reader	
handle	the IpatchFileHandle	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_sf2_reader_select_preset ()

```
gboolean
ags_ipatch_sf2_reader_select_preset (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                     guint preset_index);
```

Select preset.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
preset_index	the preset index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_sf2_reader_select_instrument ()

```
gboolean
ags_ipatch_sf2_reader_select_instrument
                                (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                 guint instrument_index);
```

Select instrument.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
instrument_index	the instrument index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_sf2_reader_select_sample ()

```
gboolean
ags_ipatch_sf2_reader_select_sample (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                     guint sample_index);
```

Select sample.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
sample_index	the sample index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_sf2_reader_get_preset_all ()

```
gchar~**
ags_ipatch_sf2_reader_get_preset_all (AgsIpatchSF2Reader *ipatch_sf2_reader);
```

Get all preset names.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	

Returns

the string vector cotaining preset names or NULL on failure.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_sf2_reader_get_instrument_all ()

```
gchar~**
ags_ipatch_sf2_reader_get_instrument_all
                                (AgsIpatchSF2Reader *ipatch_sf2_reader);
```

Get all instrument names.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader

Returns

the string vector cotaining instrument names or NULL on failure.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_sf2_reader_get_sample_all ()

```
gchar~**
ags_ipatch_sf2_reader_get_sample_all (AgsIpatchSF2Reader *ipatch_sf2_reader);
```

Get all sample names.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader

Returns

the string vector cotaining sample names or NULL on failure.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_sf2_reader_get_instrument_by_preset_index ()

```
gchar~**
ags_ipatch_sf2_reader_get_instrument_by_preset_index
                                (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                 guint preset_index);
```

Get instrument by preset index.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader
preset_index	the preset index

Returns

the string vector cotaining instrument names or NULL on failure.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_sf2_reader_get_sample_by_preset_index ()

```
gchar~**
ags_ipatch_sf2_reader_get_sample_by_preset_index
                                (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                 guint preset_index);
```

Get sample by preset index.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
preset_index	the preset index	

Returns

the string vector cotaining sample names or NULL on failure.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_sf2_reader_get_sample_by_preset_and_instrument_index ()

```
gchar~**
ags_ipatch_sf2_reader_get_sample_by_preset_and_instrument_index
                                (AgsIpatchSF2Reader *ipatch_sf2_reader,
                                 guint preset_index,
                                 guint instrument_index);
```

Get sample by preset and instrument index.

Parameters

ipatch_sf2_reader	the AgsIpatchSF2Reader	
preset_index	the preset index	
instrument_index	the instrument index	

Returns

the string vector cotaining sample names or NULL on failure.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_sf2_reader_new ()

```
AgsIpatchSF2Reader~*
ags_ipatch_sf2_reader_new (AgsIpatch *ipatch);
```

Creates a new instance of AgsIpatchSF2Reader.

Parameters

ipatch	the AgsIpatch

Returns

the new AgsIpatchSF2Reader.

Since: 3.0.0

AGS_IPATCH_SF2_READER()

```
#define AGS_IPATCH_SF2_READER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_IPATCH_SF2_READER, AgsIpatchSF2Reader))
```

AGS_IPATCH_SF2_READER_CLASS()

```
#define AGS_IPATCH_SF2_READER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_IPATCH_SF2_READER, AgsIpatchSF2ReaderClass))
```

AGS_IPATCH_SF2_READER_GET_CLASS()

```
#define AGS_IPATCH_SF2_READER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_IPATCH_SF2_READER, AgsIpatchSF2ReaderClass))
```

AGS_IS_IPATCH_SF2_READER()

```
#define AGS_IS_IPATCH_SF2_READER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_IPATCH_SF2_READER))
```

AGS_IS_IPATCH_SF2_READER_CLASS()

```
#define AGS_IS_IPATCH_SF2_READER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_IPATCH_SF2_READER))
```

ags_ipatch_sf2_reader_get_type ()

```
GType
ags_ipatch_sf2_reader_get_type ();
```

Types and Values

enum AgsSF2Level

Enum values to describe the different levels of a Soundfont2 file.

Members

AGS_SF2_FILENAME	filename
AGS_SF2_PHDR	preset header
AGS_SF2_IHDR	instrument header
AGS_SF2_SHDR	sample header

AGS_TYPE_IPATCH_SF2_READER

```
#define AGS_TYPE_IPATCH_SF2_READER                (ags_ipatch_sf2_reader_get_type())
```

struct AgsIpatchSF2Reader

```
struct AgsIpatchSF2Reader;
```

struct AgsIpatchSF2ReaderClass

```
struct AgsIpatchSF2ReaderClass {
  GObjectClass gobject;
};
```

Property Details

The "ipatch" property

```
  "ipatch"                   AgsIpatch~*
```

The assigned AgsIpatch

Owner: AgsIpatchSF2Reader

Flags: Read / Write

Since: 3.0.0

18.10 AgsIpatchDLS2Reader

AgsIpatchDLS2Reader — interfacing DLS2 related API of libinstpatch

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_IPATCH_DLS2_READER_GET_OBJ_MUTEX()
gboolean	ags_ipatch_dls2_reader_test_flags ()
void	ags_ipatch_dls2_reader_set_flags ()
void	ags_ipatch_dls2_reader_unset_flags ()
gboolean	ags_ipatch_dls2_reader_load ()
gboolean	ags_ipatch_dls2_reader_select_instrument ()
gboolean	ags_ipatch_dls2_reader_select_sample ()
gchar **	ags_ipatch_dls2_reader_get_instrument_all ()
gchar **	ags_ipatch_dls2_reader_get_sample_all ()
gchar **	ags_ipatch_dls2_reader_get_sample_by_instrument_index ()
AgsIpatchDLS2Reader *	ags_ipatch_dls2_reader_new ()
#define	AGS_IPATCH_DLS2_READER()
#define	AGS_IPATCH_DLS2_READER_CLASS()
#define	AGS_IPATCH_DLS2_READER_GET_CLASS()
#define	AGS_IS_IPATCH_DLS2_READER()
#define	AGS_IS_IPATCH_DLS2_READER_CLASS()
GType	ags_ipatch_dls2_reader_get_type ()

Properties

AgsIpatch *	ipatch	Read / Write

Types and Values

enum	AgsDLS2Level
#define	AGS_TYPE_IPATCH_DLS2_READER
struct	AgsIpatchDLS2Reader
struct	AgsIpatchDLS2ReaderClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsIpatchDLS2Reader
```

Implemented Interfaces

AgsIpatchDLS2Reader implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_ipatch_dls2_reader.h>
```

Description

AgsIpatchDLS2Reader is the base object to ineract with DLS2 related API.

Functions

AGS_IPATCH_DLS2_READER_GET_OBJ_MUTEX()

```
#define AGS_IPATCH_DLS2_READER_GET_OBJ_MUTEX(obj) (&(((AgsIpatchDLS2Reader *) obj)-> ↩
    obj_mutex))
```

ags_ipatch_dls2_reader_test_flags ()

```
gboolean
ags_ipatch_dls2_reader_test_flags (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                                   guint flags);
```

Test *flags* to be set on *ipatch_dls2_reader* .

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_ipatch_dls2_reader_set_flags ()

```
void
ags_ipatch_dls2_reader_set_flags (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                                  guint flags);
```

Enable a feature of *ipatch_dls2_reader* .

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
flags	see AgsIpatchDLS2ReaderFlags	

Since: 3.0.0

ags_ipatch_dls2_reader_unset_flags ()

```
void
ags_ipatch_dls2_reader_unset_flags (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                                    guint flags);
```

Disable a feature of *ipatch_dls2_reader* .

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
flags	see AgsIpatchDLS2ReaderFlags	

Since: 3.0.0

ags_ipatch_dls2_reader_load ()

```
gboolean
ags_ipatch_dls2_reader_load (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                             IpatchFileHandle *handle);
```

Load DLS2 file.

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
handle	the IpatchFileHandle	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_dls2_reader_select_instrument ()

```
gboolean
ags_ipatch_dls2_reader_select_instrument
                                (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                                 guint instrument_index);
```

Select instrument.

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
instrument_index	the instrument index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_dls2_reader_select_sample ()

```
gboolean
ags_ipatch_dls2_reader_select_sample (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                                      guint sample_index);
```

Select sample.

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
sample_index	the sample index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_ipatch_dls2_reader_get_instrument_all ()

```
gchar~**
ags_ipatch_dls2_reader_get_instrument_all
                                (AgsIpatchDLS2Reader *ipatch_dls2_reader);
```

Get instrument all.

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	

Returns

the string vector containing instrument.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_dls2_reader_get_sample_all ()

```
gchar~**
ags_ipatch_dls2_reader_get_sample_all (AgsIpatchDLS2Reader *ipatch_dls2_reader);
```

Get sample all.

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	

Returns

the string vector containing sample.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_dls2_reader_get_sample_by_instrument_index ()

```
gchar~**
ags_ipatch_dls2_reader_get_sample_by_instrument_index
                                (AgsIpatchDLS2Reader *ipatch_dls2_reader,
                                 guint instrument_index);
```

Get sample by instrument index.

Parameters

ipatch_dls2_reader	the AgsIpatchDLS2Reader	
instrument_index	the instrument index	

Returns

the string vector containing sample.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_ipatch_dls2_reader_new ()

```
AgsIpatchDLS2Reader~*
ags_ipatch_dls2_reader_new (AgsIpatch *ipatch);
```

Creates an AgsIpatchDLS2Reader.

Parameters

ipatch	the AgsIpatch	

Returns

an empty AgsIpatchDLS2Reader.

Since: 3.0.0

AGS_IPATCH_DLS2_READER()

```
#define AGS_IPATCH_DLS2_READER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_IPATCH_DLS2_READER, AgsIpatchDLS2Reader))
```

AGS_IPATCH_DLS2_READER_CLASS()

```
#define AGS_IPATCH_DLS2_READER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_IPATCH_DLS2_READER, AgsIpatchDLS2ReaderClass))
```

AGS_IPATCH_DLS2_READER_GET_CLASS()

```
#define AGS_IPATCH_DLS2_READER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_IPATCH_DLS2_READER, AgsIpatchDLS2ReaderClass))
```

AGS_IS_IPATCH_DLS2_READER()

```
#define AGS_IS_IPATCH_DLS2_READER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_IPATCH_DLS2_READER))
```

AGS_IS_IPATCH_DLS2_READER_CLASS()

```
#define AGS_IS_IPATCH_DLS2_READER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_IPATCH_DLS2_READER))
```

ags_ipatch_dls2_reader_get_type ()

```
GType
ags_ipatch_dls2_reader_get_type ();
```

Types and Values

enum AgsDLS2Level

Enum values to describe the different levels of a DLS2 file.

Members

AGS_DLS2_FILENAME	filename
AGS_DLS2_IHDR	instrument header
AGS_DLS2_SHDR	sample header

AGS_TYPE_IPATCH_DLS2_READER

```
#define AGS_TYPE_IPATCH_DLS2_READER                (ags_ipatch_dls2_reader_get_type())
```

struct AgsIpatchDLS2Reader

```
struct AgsIpatchDLS2Reader;
```

struct AgsIpatchDLS2ReaderClass

```
struct AgsIpatchDLS2ReaderClass {
  GObjectClass gobject;
};
```

Property Details

The "ipatch" property

```
"ipatch"                    AgsIpatch~*
```

The assigned AgsIpatch

Owner: AgsIpatchDLS2Reader

Flags: Read / Write

Since: 3.0.0

18.11 AgsIpatchSample

AgsIpatchSample — interfacing Soundfont2 related API of libinstpatch

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_IPATCH_SAMPLE_GET_OBJ_MUTEX()
gboolean	ags_ipatch_sample_test_flags ()
void	ags_ipatch_sample_set_flags ()
void	ags_ipatch_sample_unset_flags ()
AgsIpatchSample *	ags_ipatch_sample_new ()
#define	AGS_IPATCH_SAMPLE()
#define	AGS_IPATCH_SAMPLE_CLASS()
#define	AGS_IPATCH_SAMPLE_GET_CLASS()
#define	AGS_IS_IPATCH_SAMPLE()
#define	AGS_IS_IPATCH_SAMPLE_CLASS()
GType	ags_ipatch_sample_get_type ()

Properties

guint	buffer-size	Read / Write
guint	format	Read / Write
GObject *	sample	Read / Write

Types and Values

#define	AGS_TYPE_IPATCH_SAMPLE
struct	AgsIpatchSample
struct	AgsIpatchSampleClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsIpatchSample
```

Implemented Interfaces

AgsIpatchSample implements AgsConnectable and AgsSoundResource.

Includes

```
#include <ags/audio/file/ags_ipatch_sample.h>
```

Description

AgsIpatchSample is the base object to ineract with Soundfont2 related API.

Functions

AGS_IPATCH_SAMPLE_GET_OBJ_MUTEX()

```
#define AGS_IPATCH_SAMPLE_GET_OBJ_MUTEX(obj) (&(((AgsIpatchSample *) obj)->obj_mutex))
```

ags_ipatch_sample_test_flags ()

```
gboolean
ags_ipatch_sample_test_flags (AgsIpatchSample *ipatch_sample,
                              guint flags);
```

Test *flags* to be set on *ipatch_sample* .

Parameters

ipatch_sample	the AgsIpatchSample	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_ipatch_sample_set_flags ()

```
void
ags_ipatch_sample_set_flags (AgsIpatchSample *ipatch_sample,
                             guint flags);
```

Enable a feature of *ipatch_sample* .

Parameters

ipatch_sample	the AgsIpatchSample	
flags	see AgsIpatchSampleFlags	

Since: 3.0.0

ags_ipatch_sample_unset_flags ()

```
void
ags_ipatch_sample_unset_flags (AgsIpatchSample *ipatch_sample,
                               guint flags);
```

Disable a feature of *ipatch_sample*.

Parameters

ipatch_sample	the AgsIpatchSample	
flags	see AgsIpatchSampleFlags	

Since: 3.0.0

ags_ipatch_sample_new ()

```
AgsIpatchSample~*
ags_ipatch_sample_new ();
```

Creates a new instance of AgsIpatchSample.

Returns

the new AgsIpatchSample.

Since: 3.0.0

AGS_IPATCH_SAMPLE()

```
#define AGS_IPATCH_SAMPLE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_IPATCH_SAMPLE, AgsIpatchSample))
```

AGS_IPATCH_SAMPLE_CLASS()

```
#define AGS_IPATCH_SAMPLE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_IPATCH_SAMPLE, AgsIpatchSampleClass))
```

AGS_IPATCH_SAMPLE_GET_CLASS()

```
#define AGS_IPATCH_SAMPLE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_IPATCH_SAMPLE, AgsIpatchSampleClass))
```

AGS_IS_IPATCH_SAMPLE()

```
#define AGS_IS_IPATCH_SAMPLE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_IPATCH_SAMPLE))
```

AGS_IS_IPATCH_SAMPLE_CLASS()

```
#define AGS_IS_IPATCH_SAMPLE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_IPATCH_SAMPLE))
```

ags_ipatch_sample_get_type ()

```
GType
ags_ipatch_sample_get_type ();
```

Types and Values

AGS_TYPE_IPATCH_SAMPLE

```
#define AGS_TYPE_IPATCH_SAMPLE                 (ags_ipatch_sample_get_type())
```

struct AgsIpatchSample

```
struct AgsIpatchSample;
```

struct AgsIpatchSampleClass

```
struct AgsIpatchSampleClass {
  GObjectClass gobject;
};
```

Property Details

The "buffer-size" property

```
  "buffer-size"              guint
```

The buffer size to be used.

Owner: AgsIpatchSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "format" property

```
  "format"                   guint
```

The format to be used.

Owner: AgsIpatchSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "sample" property

```
"sample"                    GObject~*
```

The assigned output IpatchSample.

Owner: AgsIpatchSample

Flags: Read / Write

Since: 3.0.0

18.12 AgsSndfile

AgsSndfile — Libsndfile wrapper

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SNDFILE_GET_OBJ_MUTEX()
gboolean	ags_sndfile_test_flags ()
void	ags_sndfile_set_flags ()
void	ags_sndfile_unset_flags ()
gboolean	ags_sndfile_check_suffix ()
AgsSndfile *	ags_sndfile_new ()
#define	AGS_IS_SNDFILE()
#define	AGS_IS_SNDFILE_CLASS()
#define	AGS_SNDFILE()
#define	AGS_SNDFILE_CLASS()
#define	AGS_SNDFILE_GET_CLASS()
GType	ags_sndfile_flags_get_type ()
GType	ags_sndfile_get_type ()

Properties

guint	audio-channels	Read / Write
guint	buffer-size	Read / Write
gpointer	file	Read / Write
guint	format	Read / Write

Types and Values

enum	AgsSndfileFlags
#define	AGS_TYPE_SNDFILE
struct	AgsSndfile
struct	AgsSndfileClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsSndfileFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsSndfile
```

Implemented Interfaces

AgsSndfile implements AgsConnectable and AgsSoundResource.

Includes

```
#include <ags/audio/file/ags_sndfile.h>
```

Description

AgsSndfile is the base object to ineract with libsndfile.

Functions

AGS_SNDFILE_GET_OBJ_MUTEX()

```
#define AGS_SNDFILE_GET_OBJ_MUTEX(obj) (&(((AgsSndfile *) obj)->obj_mutex))
```

ags_sndfile_test_flags ()

```
gboolean
ags_sndfile_test_flags (AgsSndfile *sndfile,
                        guint flags);
```

Test *flags* to be set on *sndfile* .

Parameters

sndfile	the AgsSndfile	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sndfile_set_flags ()

```
void
ags_sndfile_set_flags (AgsSndfile *sndfile,
                       guint flags);
```

Enable a feature of *sndfile* .

Parameters

sndfile	the AgsSndfile	
flags	see AgsSndfileFlags	

Since: 3.0.0

ags_sndfile_unset_flags ()

```
void
ags_sndfile_unset_flags (AgsSndfile *sndfile,
                         guint flags);
```

Disable a feature of *sndfile* .

Parameters

sndfile	the AgsSndfile	
flags	see AgsSndfileFlags	

Since: 3.0.0

ags_sndfile_check_suffix ()

```
gboolean
ags_sndfile_check_suffix (gchar *filename);
```

Check suffix.

Parameters

filename	the filename	

Returns

TRUE if suffix supported, else FALSE

Since: 3.6.0

ags_sndfile_new ()

```
AgsSndfile~*
ags_sndfile_new ();
```

Creates a new instance of AgsSndfile.

Returns

the new AgsSndfile.

Since: 3.0.0

AGS_IS_SNDFILE()

```
#define AGS_IS_SNDFILE(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_SNDFILE ↩
    ))
```

AGS_IS_SNDFILE_CLASS()

```
#define AGS_IS_SNDFILE_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_SNDFILE) ↩
    )
```

AGS_SNDFILE()

```
#define AGS_SNDFILE(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_SNDFILE ↩
    , AgsSndfile))
```

AGS_SNDFILE_CLASS()

```
#define AGS_SNDFILE_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_SNDFILE, ↩
     AgsSndfileClass))
```

AGS_SNDFILE_GET_CLASS()

```
#define AGS_SNDFILE_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_SNDFILE, ↩
     AgsSndfileClass))
```

ags_sndfile_flags_get_type ()

```
GType
ags_sndfile_flags_get_type ();
```

ags_sndfile_get_type ()

```
GType
ags_sndfile_get_type ();
```

Types and Values

enum AgsSndfileFlags

Enum values to control the behavior or indicate internal state of AgsSndfile by enable/disable as flags.

Members

AGS_SNDFILE_VIRTUAL	virtual IO
AGS_SNDFILE_FILL_CACHE	fill cache

AGS_TYPE_SNDFILE

```
#define AGS_TYPE_SNDFILE                (ags_sndfile_get_type())
```

struct AgsSndfile

```
struct AgsSndfile;
```

struct AgsSndfileClass

```
struct AgsSndfileClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-channels" property

```
  "audio-channels"           guint
```

The audio channels to be used.

Owner: AgsSndfile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "buffer-size" property

```
  "buffer-size"              guint
```

The buffer size to be used.

Owner: AgsSndfile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "file" property

```
"file"                    gpointer
```

The assigned output SNDFILE.

Owner: AgsSndfile

Flags: Read / Write

Since: 3.0.0

The "format" property

```
"format"                  guint
```

The format to be used.

Owner: AgsSndfile

Flags: Read / Write

Default value: 0

Since: 3.0.0

18.13 AgsSFZFile

AgsSFZFile — SFZ file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_FILE_GET_OBJ_MUTEX()
gboolean	ags_sfz_file_test_flags ()
void	ags_sfz_file_set_flags ()
void	ags_sfz_file_unset_flags ()
GList *	ags_sfz_file_get_group ()
void	ags_sfz_file_set_group ()
GList *	ags_sfz_file_get_region ()
void	ags_sfz_file_set_region ()
GList *	ags_sfz_file_get_sample ()
void	ags_sfz_file_set_sample ()
gboolean	ags_sfz_file_select_sample ()
void	ags_sfz_file_get_range ()
gboolean	ags_sfz_file_check_suffix ()
void	ags_sfz_file_parse ()
AgsSFZFile *	ags_sfz_file_new ()
#define	AGS_IS_SFZ_FILE()
#define	AGS_IS_SFZ_FILE_CLASS()
#define	AGS_SFZ_FILE()
#define	AGS_SFZ_FILE_CLASS()

#define	AGS_SFZ_FILE_GET_CLASS()
GType	ags_sfz_file_get_type ()

Properties

char *	filename	Read / Write
gpointer	group	Read / Write
char *	mode	Read / Write
gpointer	region	Read / Write
gpointer	sample	Read / Write
GObject *	soundcard	Read / Write

Types and Values

#define	AGS_SFZ_FILE_DEFAULT_CHANNELS
#define	AGS_SFZ_FILE_LOOP_MAX
#define	AGS_SFZ_FILE_READ
#define	AGS_SFZ_FILE_WRITE
enum	AgsSFZLevel
#define	AGS_TYPE_SFZ_FILE
struct	AgsSFZFile
struct	AgsSFZFileClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsSFZFile
```

Implemented Interfaces

AgsSFZFile implements AgsConnectable and AgsSoundContainer.

Includes

```
#include <ags/audio/file/ags_sfz_file.h>
```

Description

AgsSFZFile is the base object to ineract with SFZ files.

Functions

AGS_SFZ_FILE_GET_OBJ_MUTEX()

```
#define AGS_SFZ_FILE_GET_OBJ_MUTEX(obj) (&(((AgsSFZFile *) obj)->obj_mutex))
```

ags_sfz_file_test_flags ()

```
gboolean
ags_sfz_file_test_flags (AgsSFZFile *sfz_file,
                         guint flags);
```

Test *flags* to be set on *sfz_file* .

Parameters

sfz_file	the AgsSFZFile	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sfz_file_set_flags ()

```
void
ags_sfz_file_set_flags (AgsSFZFile *sfz_file,
                        guint flags);
```

Enable a feature of *sfz_file* .

Parameters

sfz_file	the AgsSFZFile	
flags	see AgsSFZFileFlags	

Since: 3.0.0

ags_sfz_file_unset_flags ()

```
void
ags_sfz_file_unset_flags (AgsSFZFile *sfz_file,
                          guint flags);
```

Disable a feature of *sfz_file* .

Parameters

sfz_file	the AgsSFZFile	
flags	see AgsSFZFileFlags	

Since: 3.0.0

ags_sfz_file_get_group ()

```
GList~*
ags_sfz_file_get_group (AgsSFZFile *sfz_file);
```

Get the groups of *sfz_file* .

Parameters

sfz_file	the AgsSFZFile	

Returns

the GList containing AgsSFZGroup.

[element-type AgsAudio.SFZGroup][transfer full]

Since: 3.17.0

ags_sfz_file_set_group ()

```
void
ags_sfz_file_set_group (AgsSFZFile *sfz_file,
                        GList *group);
```

Set the group field of *sfz_file*

Parameters

sfz_file	the AgsSFZFile	
group	the GList containing AgsSFZGroup.	*[element-type AgsAudio.SFZGroup][transfer none]*

Since: 3.17.0

ags_sfz_file_get_region ()

```
GList~*
ags_sfz_file_get_region (AgsSFZFile *sfz_file);
```

Get the regions of *sfz_file* .

Parameters

sfz_file	the AgsSFZFile	

Returns

the GList containing AgsSFZRegion.

[element-type AgsAudio.SFZRegion][transfer full]

Since: 3.17.0

ags_sfz_file_set_region ()

```
void
ags_sfz_file_set_region (AgsSFZFile *sfz_file,
                         GList *region);
```

Set the region field of *sfz_file*

Parameters

sfz_file	the AgsSFZFile	
region	the GList containing AgsSFZRegion.	*[element-type AgsAudio.SFZRegion][transfer none]*

Since: 3.17.0

ags_sfz_file_get_sample ()

```
GList~*
ags_sfz_file_get_sample (AgsSFZFile *sfz_file);
```

Get the samples of *sfz_file* .

Parameters

sfz_file	the AgsSFZFile	

Returns

the GList containing AgsSFZSample.

[element-type AgsAudio.SFZSample][transfer full]

Since: 3.17.0

ags_sfz_file_set_sample ()

```
void
ags_sfz_file_set_sample (AgsSFZFile *sfz_file,
                         GList *sample);
```

Set the sample field of *sfz_file*

Parameters

sfz_file	the AgsSFZFile	
sample	the GList containing AgsSFZSample.	*[element-type AgsAudio.SFZSample][transfer none]*

Since: 3.17.0

ags_sfz_file_select_sample ()

```
gboolean
ags_sfz_file_select_sample (AgsSFZFile *sfz_file,
                            guint sample_index);
```

Select sample.

Parameters

sfz_file	the AgsSFZFile	
sample_index	the sample index	

Returns

TRUE on success, else FALSE on failure

Since: 3.0.0

ags_sfz_file_get_range ()

```
void
ags_sfz_file_get_range (AgsSFZFile *sfz_file,
                        glong *hikey,
                        glong *lokey);
```

Get range of *sfz_file* , set return location *hikey* and *lokey* .

Parameters

sfz_file	the AgsSFZFile	
hikey	the return location of key high.	*[out]*
lokey	the return location of key low.	*[out]*

Since: 3.0.0

ags_sfz_file_check_suffix ()

```
gboolean
ags_sfz_file_check_suffix (gchar *filename);
```

Check *filename* 's suffix to be supported.

Parameters

filename	the filename	

Returns

TRUE if supported, else FALSE

Since: 3.0.0

ags_sfz_file_parse ()

```
void
ags_sfz_file_parse (AgsSFZFile *sfz_file);
```

Parse *sfz_file*.

Parameters

sfz_file	the AgsSFZFile

Since: 3.0.0

ags_sfz_file_new ()

```
AgsSFZFile~*
ags_sfz_file_new ();
```

Creates an AgsSFZFile.

Returns

an empty AgsSFZFile.

Since: 3.0.0

AGS_IS_SFZ_FILE()

```
#define AGS_IS_SFZ_FILE(obj)            (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SFZ_FILE))
```

AGS_IS_SFZ_FILE_CLASS()

```
#define AGS_IS_SFZ_FILE_CLASS(class)    (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SFZ_FILE))
```

AGS_SFZ_FILE()

```
#define AGS_SFZ_FILE(obj)               (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_FILE, AgsSFZFile))
```

AGS_SFZ_FILE_CLASS()

```
#define AGS_SFZ_FILE_CLASS(class)       (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_FILE, AgsSFZFileClass))
```

AGS_SFZ_FILE_GET_CLASS()

```
#define AGS_SFZ_FILE_GET_CLASS(obj)     (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SFZ_FILE, AgsSFZFileClass))
```

ags_sfz_file_get_type ()

```
GType
ags_sfz_file_get_type ();
```

Types and Values

AGS_SFZ_FILE_DEFAULT_CHANNELS

```
#define AGS_SFZ_FILE_DEFAULT_CHANNELS (2)
```

AGS_SFZ_FILE_LOOP_MAX

```
#define AGS_SFZ_FILE_LOOP_MAX (4294967296)
```

AGS_SFZ_FILE_READ

```
#define AGS_SFZ_FILE_READ "r"
```

AGS_SFZ_FILE_WRITE

```
#define AGS_SFZ_FILE_WRITE "w"
```

enum AgsSFZLevel

Enum values to describe the different levels of a SFZ file.

Members

AGS_SFZ_LEVEL_FILENAME	filename
AGS_SFZ_LEVEL_SAMPLE	sample

AGS_TYPE_SFZ_FILE

```
#define AGS_TYPE_SFZ_FILE                (ags_sfz_file_get_type())
```

struct AgsSFZFile

```
struct AgsSFZFile;
```

struct AgsSFZFileClass

```
struct AgsSFZFileClass {
  GObjectClass gobject;
};
```

Property Details

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsSFZFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "group" property

```
"group"                     gpointer
```

The containing groups.

[transfer full]

Owner: AgsSFZFile

Flags: Read / Write

Since: 3.0.0

The "mode" property

```
"mode"                      char~*
```

The assigned mode.

Owner: AgsSFZFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "region" property

```
"region"                    gpointer
```

The containing regions.

[transfer full]

Owner: AgsSFZFile

Flags: Read / Write

Since: 3.0.0

The "sample" property

```
"sample"                    gpointer
```

The containing samples.

[transfer full]

Owner: AgsSFZFile

Flags: Read / Write

Since: 3.0.0

The "soundcard" property

```
"soundcard"                 GObject~*
```

The assigned soundcard.

Owner: AgsSFZFile

Flags: Read / Write

Since: 3.0.0

18.14 AgsSFZGroup

AgsSFZGroup — interfacing SFZ groups

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_GROUP_GET_OBJ_MUTEX()
gboolean	ags_sfz_group_test_flags ()
void	ags_sfz_group_set_flags ()
void	ags_sfz_group_unset_flags ()
GList *	ags_sfz_group_get_region ()
void	ags_sfz_group_set_region ()
GObject *	ags_sfz_group_get_sample ()
void	ags_sfz_group_set_sample ()
GList *	ags_sfz_group_get_control ()
void	ags_sfz_group_insert_control ()
gchar *	ags_sfz_group_lookup_control ()
AgsSFZGroup *	ags_sfz_group_new ()
#define	AGS_IS_SFZ_GROUP()
#define	AGS_IS_SFZ_GROUP_CLASS()
#define	AGS_SFZ_GROUP()
#define	AGS_SFZ_GROUP_CLASS()
#define	AGS_SFZ_GROUP_GET_CLASS()
GType	ags_sfz_group_get_type ()

Properties

int	nth-group	Read / Write
gpointer	region	Read / Write
AgsSFZSample *	sample	Read / Write

Types and Values

#define	AGS_TYPE_SFZ_GROUP
struct	AgsSFZGroup
struct	AgsSFZGroupClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsSFZGroup
```

Implemented Interfaces

AgsSFZGroup implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_sfz_group.h>
```

Description

AgsSFZGroup is the base object to ineract with SFZ groups.

Functions

AGS_SFZ_GROUP_GET_OBJ_MUTEX()

```
#define AGS_SFZ_GROUP_GET_OBJ_MUTEX(obj) (&(((AgsSFZGroup *) obj)->obj_mutex))
```

ags_sfz_group_test_flags ()

```
gboolean
ags_sfz_group_test_flags (AgsSFZGroup *sfz_group,
                          guint flags);
```

Test *flags* to be set on *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sfz_group_set_flags ()

```
void
ags_sfz_group_set_flags (AgsSFZGroup *sfz_group,
                         guint flags);
```

Enable a feature of *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup	
flags	see AgsSFZGroupFlags	

Since: 3.0.0

ags_sfz_group_unset_flags ()

```
void
ags_sfz_group_unset_flags (AgsSFZGroup *sfz_group,
                           guint flags);
```

Disable a feature of *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup	
flags	see AgsSFZGroupFlags	

Since: 3.0.0

ags_sfz_group_get_region ()

```
GList~*
ags_sfz_group_get_region (AgsSFZGroup *sfz_group);
```

Get the regions of *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup	

Returns

the GList containing AgsSFZRegion.

[element-type AgsAudio.SFZRegion][transfer full]

Since: 3.17.0

ags_sfz_group_set_region ()

```
void
ags_sfz_group_set_region (AgsSFZGroup *sfz_group,
                          GList *region);
```

Set the region field of *sfz_group*

Parameters

sfz_group	the AgsSFZGroup	
region	the GList containing AgsSFZRegion.	*[element-type AgsAudio.SFZRegion][transfer none]*

Since: 3.17.0

ags_sfz_group_get_sample ()

```
GObject~*
ags_sfz_group_get_sample (AgsSFZGroup *sfz_group);
```

Get the samples of *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup	

Returns

the AgsSFZSample.

[transfer full]

Since: 3.17.0

ags_sfz_group_set_sample ()

```
void
ags_sfz_group_set_sample (AgsSFZGroup *sfz_group,
                          GObject *sample);
```

Set the sample field of *sfz_group*

Parameters

sfz_group	the AgsSFZGroup	
sample	the AgsSFZSample.	*[transfer none]*

Since: 3.17.0

ags_sfz_group_get_control ()

```
GList~*
ags_sfz_group_get_control (AgsSFZGroup *sfz_group);
```

Get all control of *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup

Returns

the GList containing controls as string.

[element-type utf8][transfer container]

Since: 3.17.0

ags_sfz_group_insert_control ()

```
void
ags_sfz_group_insert_control (AgsSFZGroup *sfz_group,
                              gchar *key,
                              gchar *value);
```

Insert control specified by *key* and *value* to *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup
key	the key
value	the value

Since: 3.0.0

ags_sfz_group_lookup_control ()

```
gchar~*
ags_sfz_group_lookup_control (AgsSFZGroup *sfz_group,
                              gchar *key);
```

Lookup control specified by *key* of *sfz_group* .

Parameters

sfz_group	the AgsSFZGroup
key	the key

Returns

the matching value of key.

[transfer full]

Since: 3.0.0

ags_sfz_group_new ()

```
AgsSFZGroup~*
ags_sfz_group_new ();
```

Creates a new instance of AgsSFZGroup.

Returns

the new AgsSFZGroup.

Since: 3.0.0

AGS_IS_SFZ_GROUP()

```
#define AGS_IS_SFZ_GROUP(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SFZ_GROUP))
```

AGS_IS_SFZ_GROUP_CLASS()

```
#define AGS_IS_SFZ_GROUP_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SFZ_GROUP))
```

AGS_SFZ_GROUP()

```
#define AGS_SFZ_GROUP(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_GROUP, AgsSFZGroup))
```

AGS_SFZ_GROUP_CLASS()

```
#define AGS_SFZ_GROUP_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_GROUP, AgsSFZGroupClass))
```

AGS_SFZ_GROUP_GET_CLASS()

```
#define AGS_SFZ_GROUP_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SFZ_GROUP, AgsSFZGroupClass))
```

ags_sfz_group_get_type ()

```
GType
ags_sfz_group_get_type ();
```

Types and Values

AGS_TYPE_SFZ_GROUP

```
#define AGS_TYPE_SFZ_GROUP                 (ags_sfz_group_get_type())
```

struct AgsSFZGroup

```
struct AgsSFZGroup;
```

struct AgsSFZGroupClass

```
struct AgsSFZGroupClass {
  GObjectClass gobject;
};
```

Property Details

The "nth-group" property

```
  "nth-group"                 int
```

The nth group.

Owner: AgsSFZGroup

Flags: Read / Write

Allowed values: >= -1

Default value: -1

Since: 3.0.0

The "region" property

```
  "region"                    gpointer
```

The region assigned with.

[transfer full]

Owner: AgsSFZGroup

Flags: Read / Write

Since: 3.0.0

The "sample" property

```
  "sample"                    AgsSFZSample~*
```

The sample assigned with.

Owner: AgsSFZGroup

Flags: Read / Write

Since: 3.0.0

18.15 AgsSFZRegion

AgsSFZRegion — interfacing SFZ regions

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_REGION_GET_OBJ_MUTEX()
gboolean	ags_sfz_region_test_flags ()
void	ags_sfz_region_set_flags ()
void	ags_sfz_region_unset_flags ()
GObject *	ags_sfz_region_get_group ()
void	ags_sfz_region_set_group ()
GObject *	ags_sfz_region_get_sample ()
void	ags_sfz_region_set_sample ()
GList *	ags_sfz_region_get_control ()
void	ags_sfz_region_insert_control ()
gchar *	ags_sfz_region_lookup_control ()
AgsSFZRegion *	ags_sfz_region_new ()
#define	AGS_IS_SFZ_REGION()
#define	AGS_IS_SFZ_REGION_CLASS()
#define	AGS_SFZ_REGION()
#define	AGS_SFZ_REGION_CLASS()
#define	AGS_SFZ_REGION_GET_CLASS()
GType	ags_sfz_region_get_type ()

Properties

AgsSFZGroup *	group	Read / Write
int	nth-region	Read / Write
AgsSFZSample *	sample	Read / Write

Types and Values

#define	AGS_TYPE_SFZ_REGION
struct	AgsSFZRegion
struct	AgsSFZRegionClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsSFZRegion
```

Implemented Interfaces

AgsSFZRegion implements AgsConnectable.

Includes

```
#include <ags/audio/file/ags_sfz_region.h>
```

Description

AgsSFZRegion is the base object to ineract with SFZ regions.

Functions

AGS_SFZ_REGION_GET_OBJ_MUTEX()

```
#define AGS_SFZ_REGION_GET_OBJ_MUTEX(obj) (&(((AgsSFZRegion *) obj)->obj_mutex))
```

ags_sfz_region_test_flags ()

```
gboolean
ags_sfz_region_test_flags (AgsSFZRegion *sfz_region,
                           guint flags);
```

Test *flags* to be set on *sfz_region* .

Parameters

sfz_region	the AgsSFZRegion	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sfz_region_set_flags ()

```
void
ags_sfz_region_set_flags (AgsSFZRegion *sfz_region,
                          guint flags);
```

Enable a feature of *sfz_region* .

Parameters

sfz_region	the AgsSFZRegion	
flags	see AgsSFZRegionFlags	

Since: 3.0.0

ags_sfz_region_unset_flags ()

```
void
ags_sfz_region_unset_flags (AgsSFZRegion *sfz_region,
                            guint flags);
```

Disable a feature of *sfz_region* .

Parameters

sfz_region	the AgsSFZRegion	
flags	see AgsSFZRegionFlags	

Since: 3.0.0

ags_sfz_region_get_group ()

```
GObject~*
ags_sfz_region_get_group (AgsSFZRegion *sfz_region);
```

Get the groups of *sfz_region* .

Parameters

sfz_region	the AgsSFZRegion	

Returns

the AgsSFZGroup.

[transfer full]

Since: 3.17.0

ags_sfz_region_set_group ()

```
void
ags_sfz_region_set_group (AgsSFZRegion *sfz_region,
                          GObject *group);
```

Set the group field of *sfz_region*

Parameters

sfz_region	the AgsSFZRegion	
group	the AgsSFZGroup.	*[transfer none]*

Since: 3.17.0

ags_sfz_region_get_sample ()

```
GObject~*
ags_sfz_region_get_sample (AgsSFZRegion *sfz_region);
```

Get the samples of *sfz_region* .

Parameters

sfz_region	the AgsSFZRegion	

Returns

the AgsSFZSample.

[transfer full]

Since: 3.17.0

ags_sfz_region_set_sample ()

```
void
ags_sfz_region_set_sample (AgsSFZRegion *sfz_region,
                           GObject *sample);
```

Set the sample field of `sfz_region`

Parameters

sfz_region	the AgsSFZRegion	
sample	the AgsSFZSample.	*[transfer none]*

Since: 3.17.0

ags_sfz_region_get_control ()

```
GList~*
ags_sfz_region_get_control (AgsSFZRegion *sfz_region);
```

Get all control of `sfz_region` .

Parameters

sfz_region	the AgsSFZRegion	

Returns

the GList containing controls as string.

[element-type utf8][transfer container]

Since: 3.17.0

ags_sfz_region_insert_control ()

```
void
ags_sfz_region_insert_control (AgsSFZRegion *sfz_region,
                               gchar *key,
                               gchar *value);
```

Insert control specified by `key` and `value` to `sfz_region` .

Parameters

sfz_region	the AgsSFZRegion	
key	the key	
value	the value	

Since: 3.0.0

ags_sfz_region_lookup_control ()

```
gchar~*
ags_sfz_region_lookup_control (AgsSFZRegion *sfz_region,
                               gchar *key);
```

Lookup control specified by *key* of *sfz_region* .

Parameters

sfz_region	the AgsSFZRegion	
key	the key	

Returns

the matching value of key.

[transfer full]

Since: 3.0.0

ags_sfz_region_new ()

```
AgsSFZRegion~*
ags_sfz_region_new ();
```

Creates a new instance of AgsSFZRegion.

Returns

the new AgsSFZRegion.

Since: 3.0.0

AGS_IS_SFZ_REGION()

```
#define AGS_IS_SFZ_REGION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SFZ_REGION))
```

AGS_IS_SFZ_REGION_CLASS()

```
#define AGS_IS_SFZ_REGION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SFZ_REGION))
```

AGS_SFZ_REGION()

```
#define AGS_SFZ_REGION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_REGION, AgsSFZRegion))
```

AGS_SFZ_REGION_CLASS()

```
#define AGS_SFZ_REGION_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_REGION, AgsSFZRegionClass))
```

AGS_SFZ_REGION_GET_CLASS()

```
#define AGS_SFZ_REGION_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SFZ_REGION, AgsSFZRegionClass))
```

ags_sfz_region_get_type ()

```
GType
ags_sfz_region_get_type ();
```

Types and Values

AGS_TYPE_SFZ_REGION

```
#define AGS_TYPE_SFZ_REGION                 (ags_sfz_region_get_type())
```

struct AgsSFZRegion

```
struct AgsSFZRegion;
```

struct AgsSFZRegionClass

```
struct AgsSFZRegionClass {
  GObjectClass gobject;
};
```

Property Details

The "group" property

```
  "group"                     AgsSFZGroup~*
```

The group assigned with.

Owner: AgsSFZRegion

Flags: Read / Write

Since: 3.0.0

The "nth-region" property

```
"nth-region"                int
```

The nth region.

Owner: AgsSFZRegion

Flags: Read / Write

Allowed values: >= -1

Default value: -1

Since: 3.0.0

The "sample" property

```
"sample"                    AgsSFZSample~*
```

The sample assigned with.

Owner: AgsSFZRegion

Flags: Read / Write

Since: 3.0.0

18.16 AgsSFZSample

AgsSFZSample — interfacing SFZ samples

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SFZ_SAMPLE_GET_OBJ_MUTEX()
gboolean	ags_sfz_sample_test_flags ()
void	ags_sfz_sample_set_flags ()
void	ags_sfz_sample_unset_flags ()
GObject *	ags_sfz_sample_get_group ()
void	ags_sfz_sample_set_group ()
GObject *	ags_sfz_sample_get_region ()
void	ags_sfz_sample_set_region ()
gint	ags_sfz_sample_get_key ()
gint	ags_sfz_sample_get_hikey ()
gint	ags_sfz_sample_get_lokey ()
gint	ags_sfz_sample_get_pitch_keycenter ()
guint	ags_sfz_sample_get_loop_mode ()
guint	ags_sfz_sample_get_loop_start ()
guint	ags_sfz_sample_get_loop_end ()
AgsSFZSample *	ags_sfz_sample_new ()
#define	AGS_IS_SFZ_SAMPLE()

#define	AGS_IS_SFZ_SAMPLE_CLASS()
#define	AGS_SFZ_SAMPLE()
#define	AGS_SFZ_SAMPLE_CLASS()
#define	AGS_SFZ_SAMPLE_GET_CLASS()
GType	ags_sfz_sample_get_type ()

Properties

guint	audio-channels	Read / Write
guint	buffer-size	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
AgsSFZGroup *	group	Read / Write
guint	loop-end	Read / Write
guint	loop-start	Read / Write
AgsSFZRegion *	region	Read / Write

Types and Values

enum	AgsSFZSampleFlags
enum	AgsSFZSampleLoopMode
#define	AGS_TYPE_SFZ_SAMPLE
struct	AgsSFZSample
struct	AgsSFZSampleClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsSFZSample
```

Implemented Interfaces

AgsSFZSample implements AgsConnectable and AgsSoundResource.

Includes

```
#include <ags/audio/file/ags_sfz_sample.h>
```

Description

AgsSFZSample is the base object to ineract with SFZ samples.

Functions

AGS_SFZ_SAMPLE_GET_OBJ_MUTEX()

```
#define AGS_SFZ_SAMPLE_GET_OBJ_MUTEX(obj) (&(((AgsSFZSample *) obj)->obj_mutex))
```

ags_sfz_sample_test_flags ()

```
gboolean
ags_sfz_sample_test_flags (AgsSFZSample *sfz_sample,
                           guint flags);
```

Test *flags* to be set on *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_sfz_sample_set_flags ()

```
void
ags_sfz_sample_set_flags (AgsSFZSample *sfz_sample,
                          guint flags);
```

Enable a feature of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	
flags	see AgsSFZSampleFlags	

Since: 3.0.0

ags_sfz_sample_unset_flags ()

```
void
ags_sfz_sample_unset_flags (AgsSFZSample *sfz_sample,
                            guint flags);
```

Disable a feature of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	
flags	see AgsSFZSampleFlags	

Since: 3.0.0

ags_sfz_sample_get_group ()

```
GObject~*
ags_sfz_sample_get_group (AgsSFZSample *sfz_sample);
```

Get the groups of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	

Returns

the AgsSFZGroup.

[transfer full]

Since: 3.17.0

ags_sfz_sample_set_group ()

```
void
ags_sfz_sample_set_group (AgsSFZSample *sfz_sample,
                          GObject *group);
```

Set the group field of *sfz_sample*

Parameters

sfz_sample	the AgsSFZSample	
group	the AgsSFZGroup.	*[transfer none]*

Since: 3.17.0

ags_sfz_sample_get_region ()

```
GObject~*
ags_sfz_sample_get_region (AgsSFZSample *sfz_sample);
```

Get the regions of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	

Returns

the AgsSFZRegion.

[transfer full]

Since: 3.17.0

ags_sfz_sample_set_region ()

```
void
ags_sfz_sample_set_region (AgsSFZSample *sfz_sample,
                           GObject *region);
```

Set the region field of *sfz_sample*

Parameters

sfz_sample	the AgsSFZSample	
region	the AgsSFZRegion.	*[transfer none]*

Since: 3.17.0

ags_sfz_sample_get_key ()

```
gint
ags_sfz_sample_get_key (AgsSFZSample *sfz_sample);
```

Get key of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	

Returns

the key

Since: 3.7.3

ags_sfz_sample_get_hikey ()

```
gint
ags_sfz_sample_get_hikey (AgsSFZSample *sfz_sample);
```

Get high key of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample	

Returns

the key

Since: 3.7.3

ags_sfz_sample_get_lokey ()

```
gint
ags_sfz_sample_get_lokey (AgsSFZSample *sfz_sample);
```

Get low key of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample

Returns

the key

Since: 3.7.3

ags_sfz_sample_get_pitch_keycenter ()

```
gint
ags_sfz_sample_get_pitch_keycenter (AgsSFZSample *sfz_sample);
```

Get pitch key-center of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample

Returns

the key

Since: 3.7.3

ags_sfz_sample_get_loop_mode ()

```
guint
ags_sfz_sample_get_loop_mode (AgsSFZSample *sfz_sample);
```

Get key of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample

Returns

the key

Since: 3.7.3

ags_sfz_sample_get_loop_start ()

```
guint
ags_sfz_sample_get_loop_start (AgsSFZSample *sfz_sample);
```

Get key of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample

Returns

the key

Since: 3.7.3

ags_sfz_sample_get_loop_end ()

```
guint
ags_sfz_sample_get_loop_end (AgsSFZSample *sfz_sample);
```

Get key of *sfz_sample* .

Parameters

sfz_sample	the AgsSFZSample

Returns

the key

Since: 3.7.3

ags_sfz_sample_new ()

```
AgsSFZSample~*
ags_sfz_sample_new ();
```

Creates a new instance of AgsSFZSample.

Returns

the new AgsSFZSample.

Since: 3.0.0

AGS_IS_SFZ_SAMPLE()

```
#define AGS_IS_SFZ_SAMPLE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SFZ_SAMPLE))
```

AGS_IS_SFZ_SAMPLE_CLASS()

```
#define AGS_IS_SFZ_SAMPLE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SFZ_SAMPLE))
```

AGS_SFZ_SAMPLE()

```
#define AGS_SFZ_SAMPLE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SFZ_SAMPLE, AgsSFZSample))
```

AGS_SFZ_SAMPLE_CLASS()

```
#define AGS_SFZ_SAMPLE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SFZ_SAMPLE, AgsSFZSampleClass))
```

AGS_SFZ_SAMPLE_GET_CLASS()

```
#define AGS_SFZ_SAMPLE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_SFZ_SAMPLE, AgsSFZSampleClass))
```

ags_sfz_sample_get_type ()

```
GType
ags_sfz_sample_get_type ();
```

Types and Values

enum AgsSFZSampleFlags

Enum values to control the behavior or indicate internal state of AgsSFZSample by enable/disable as flags.

Members

AGS_SFZ_SAMPLE_ADDED_TO_REGISTRY	the sfz sample was added to registry, see AgsConnectable::add_to_registry()

AGS_SFZ_SAMPLE_CONNECTED	indicates the sfz sample was connected by calling AgsConnectable::connect()

enum AgsSFZSampleLoopMode

Enum values to control the behavior or indicate internal state of AgsSFZSample by enable/disable as flags.

Members

AGS_SFZ_SAMPLE_LOOP_ONE_SHOT	one shot
AGS_SFZ_SAMPLE_LOOP_CONTINUOUS	loop continuous
AGS_SFZ_SAMPLE_LOOP_SUSTAIN	loop sustain

AGS_TYPE_SFZ_SAMPLE

```
#define AGS_TYPE_SFZ_SAMPLE                 (ags_sfz_sample_get_type())
```

struct AgsSFZSample

```
struct AgsSFZSample;
```

struct AgsSFZSampleClass

```
struct AgsSFZSampleClass {
  GObjectClass gobject;
};
```

Property Details

The "audio-channels" property

```
  "audio-channels"            guint
```

The audio channels to be used.

Owner: AgsSFZSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "buffer-size" property

```
"buffer-size"                guint
```

The buffer size to be used.

Owner: AgsSFZSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "filename" property

```
"filename"                   char~*
```

The filename to be used.

Owner: AgsSFZSample

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "format" property

```
"format"                     guint
```

The format to be used.

Owner: AgsSFZSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "group" property

```
"group"                      AgsSFZGroup~*
```

The group assigned with.

Owner: AgsSFZSample

Flags: Read / Write

Since: 3.0.0

The "loop-end" property

```
"loop-end"                  guint
```

The loop end to be used.

Owner: AgsSFZSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "loop-start" property

```
"loop-start"                guint
```

The loop start to be used.

Owner: AgsSFZSample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "region" property

```
"region"                    AgsSFZRegion~*
```

The region assigned with.

Owner: AgsSFZSample

Flags: Read / Write

Since: 3.0.0

18.17 AgsSoundContainer

AgsSoundContainer — read/write audio

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_sound_container_open ()
guint	ags_sound_container_get_level_count ()
guint	ags_sound_container_get_nesting_level ()
gchar *	ags_sound_container_get_level_id ()
guint	ags_sound_container_get_level_index ()
gchar **	ags_sound_container_get_sublevel_name ()

guint	ags_sound_container_level_up ()
guint	ags_sound_container_select_level_by_id ()
guint	ags_sound_container_select_level_by_index ()
GList *	ags_sound_container_get_resource_all ()
GList *	ags_sound_container_get_resource_by_name ()
GList *	ags_sound_container_get_resource_by_index ()
GList *	ags_sound_container_get_resource_current ()
void	ags_sound_container_close ()
#define	AGS_IS_SOUND_CONTAINER()
#define	AGS_IS_SOUND_CONTAINER_INTERFACE()
#define	AGS_SOUND_CONTAINER()
#define	AGS_SOUND_CONTAINER_GET_INTERFACE()
#define	AGS_SOUND_CONTAINER_INTERFACE()
GType	ags_sound_container_get_type ()

Types and Values

#define	AGS_TYPE_SOUND_CONTAINER
	AgsSoundContainer
struct	AgsSoundContainerInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsSoundContainer
```

Known Implementations

AgsSoundContainer is implemented by AgsIpatch and AgsSFZFile.

Includes

```
#include <ags/audio/file/ags_sound_container.h>
```

Description

The AgsSoundContainer interface gives you a unique access to file related IO operations.

Functions

ags_sound_container_open ()

```
gboolean
ags_sound_container_open (AgsSoundContainer *sound_container,
                          gchar *filename);
```

Open *sound_container* .

Parameters

sound_container	the AgsSoundcontainer	
filename	the filename	

Returns

TRUE if open succeeded, else FALSE

Since: 3.0.0

ags_sound_container_get_level_count ()

```
guint
ags_sound_container_get_level_count (AgsSoundContainer *sound_container);
```

Get level count.

Parameters

sound_container	the AgsSoundcontainer	

Returns

the level count

Since: 3.0.0

ags_sound_container_get_nesting_level ()

```
guint
ags_sound_container_get_nesting_level (AgsSoundContainer *sound_container);
```

Get nesting level.

Parameters

sound_container	the AgsSoundcontainer	

Returns

the nesting level

Since: 3.0.0

ags_sound_container_get_level_id ()

```
gchar~*
ags_sound_container_get_level_id (AgsSoundContainer *sound_container);
```

Get level id.

Parameters

sound_container	the AgsSoundcontainer

Returns

the level id as string

Since: 3.0.0

ags_sound_container_get_level_index ()

```
guint
ags_sound_container_get_level_index (AgsSoundContainer *sound_container);
```

Get level index.

Parameters

sound_container	the AgsSoundcontainer

Returns

the level index unsigned integer

Since: 3.0.0

ags_sound_container_get_sublevel_name ()

```
gchar~**
ags_sound_container_get_sublevel_name (AgsSoundContainer *sound_container);
```

Get sublevel names.

Parameters

sound_container	the AgsSoundcontainer

Returns

the string vector containing sublevel names.

[element-type utf8][array zero-terminated=1][transfer full]

Since: 3.0.0

ags_sound_container_level_up ()

```
guint
ags_sound_container_level_up (AgsSoundContainer *sound_container,
                              guint level_count);
```

Level up.

Parameters

sound_container	the AgsSoundcontainer	
level_count	the count of levels to go up	

Returns

the level count went up

Since: 3.0.0

ags_sound_container_select_level_by_id ()

```
guint
ags_sound_container_select_level_by_id
                                (AgsSoundContainer *sound_container,
                                 gchar *level_id);
```

Select level by id.

Parameters

sound_container	the AgsSoundcontainer	
level_id	the level id as string	

Returns

the index of level as unsigned integer

Since: 3.0.0

ags_sound_container_select_level_by_index ()

```
guint
ags_sound_container_select_level_by_index
                                (AgsSoundContainer *sound_container,
                                 guint level_index);
```

Select level by index.

Parameters

sound_container	the AgsSoundcontainer	
level_index	the level index unsigned integer	

Returns

the index of level as unsigned integer

Since: 3.0.0

ags_sound_container_get_resource_all ()

```
GList~*
ags_sound_container_get_resource_all (AgsSoundContainer *sound_container);
```

Get all resources as a GList.

Parameters

sound_container	the AgsSoundcontainer

Returns

the GList containing AgsResource.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_container_get_resource_by_name ()

```
GList~*
ags_sound_container_get_resource_by_name
                                (AgsSoundContainer *sound_container,
                                 gchar *resource_name);
```

Get resources by name as a GList.

Parameters

sound_container	the AgsSoundcontainer
resource_name	the resource name

Returns

the GList containing AgsResource.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_container_get_resource_by_index ()

```
GList~*
ags_sound_container_get_resource_by_index
                                (AgsSoundContainer *sound_container,
                                 guint resource_index);
```

Get resources by index as a GList.

Parameters

sound_container	the AgsSoundcontainer	
resource_index	the index	

Returns

the GList containing AgsResource.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_container_get_resource_current ()

```
GList~*
ags_sound_container_get_resource_current
                                (AgsSoundContainer *sound_container);
```

Get resources by index as a GList.

Parameters

sound_container	the AgsSoundcontainer	

Returns

the GList containing AgsResource.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_container_close ()

```
void
ags_sound_container_close (AgsSoundContainer *sound_container);
```

Close *sound_container*.

Parameters

sound_container	the AgsSoundcontainer	

Since: 3.0.0

AGS_IS_SOUND_CONTAINER()

```
#define AGS_IS_SOUND_CONTAINER(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SOUND_CONTAINER))
```

AGS_IS_SOUND_CONTAINER_INTERFACE()

```
#define AGS_IS_SOUND_CONTAINER_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SOUND_CONTAINER))
```

AGS_SOUND_CONTAINER()

```
#define AGS_SOUND_CONTAINER(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOUND_CONTAINER, AgsSoundContainer))
```

AGS_SOUND_CONTAINER_GET_INTERFACE()

```
#define AGS_SOUND_CONTAINER_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SOUND_CONTAINER, AgsSoundContainerInterface))
```

AGS_SOUND_CONTAINER_INTERFACE()

```
#define AGS_SOUND_CONTAINER_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SOUND_CONTAINER, AgsSoundContainerInterface))
```

ags_sound_container_get_type ()

```
GType
ags_sound_container_get_type ();
```

Types and Values

AGS_TYPE_SOUND_CONTAINER

```
#define AGS_TYPE_SOUND_CONTAINER                     (ags_sound_container_get_type())
```

AgsSoundContainer

```
typedef struct _AgsSoundContainer AgsSoundContainer;
```

struct AgsSoundContainerInterface

```
struct AgsSoundContainerInterface {
  GTypeInterface ginterface;

  gboolean (*open)(AgsSoundContainer *sound_container, gchar *filename);

  /* level information */
  guint (*get_level_count)(AgsSoundContainer *sound_container);
  guint (*get_nesting_level)(AgsSoundContainer *sound_container);

  gchar* (*get_level_id)(AgsSoundContainer *sound_container);
  guint (*get_level_index)(AgsSoundContainer *sound_container);
```

```
  gchar** (*get_sublevel_name)(AgsSoundContainer *sound_container);

  /* levels */
  guint (*level_up)(AgsSoundContainer *sound_container,
        guint level_count);
  guint (*select_level_by_id)(AgsSoundContainer *sound_container,
            gchar *level_id);
  guint (*select_level_by_index)(AgsSoundContainer *sound_container,
         guint level_index);

  /* get sound resource */
  GList* (*get_resource_all)(AgsSoundContainer *sound_container);
  GList* (*get_resource_by_name)(AgsSoundContainer *sound_container,
         gchar *resource_name);
  GList* (*get_resource_by_index)(AgsSoundContainer *sound_container,
          guint resource_index);
  GList* (*get_resource_current)(AgsSoundContainer *sound_container);

  /* close */
  void (*close)(AgsSoundContainer *sound_container);
};
```

18.18 AgsSoundResource

AgsSoundResource — read/write audio

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_sound_resource_open ()
gboolean	ags_sound_resource_rw_open ()
void	ags_sound_resource_load ()
void	ags_sound_resource_info ()
void	ags_sound_resource_set_presets ()
void	ags_sound_resource_get_presets ()
guint	ags_sound_resource_read ()
void	ags_sound_resource_write ()
void	ags_sound_resource_flush ()
void	ags_sound_resource_seek ()
GList *	ags_sound_resource_read_audio_signal ()
GList *	ags_sound_resource_read_wave ()
void	ags_sound_resource_close ()
#define	AGS_IS_SOUND_RESOURCE()
#define	AGS_IS_SOUND_RESOURCE_INTERFACE()
#define	AGS_SOUND_RESOURCE()
#define	AGS_SOUND_RESOURCE_GET_INTERFACE()
#define	AGS_SOUND_RESOURCE_INTERFACE()
GType	ags_sound_resource_get_type ()

Types and Values

#define	AGS_TYPE_SOUND_RESOURCE
	AgsSoundResource
struct	AgsSoundResourceInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsSoundResource
```

Known Implementations

AgsSoundResource is implemented by AgsGstreamerFile, AgsIpatchSample, AgsSFZSample and AgsSndfile.

Includes

```
#include <ags/audio/file/ags_sound_resource.h>
```

Description

The AgsSoundResource interface gives you a unique access to file related IO operations.

Functions

ags_sound_resource_open ()

```
gboolean
ags_sound_resource_open (AgsSoundResource *sound_resource,
                         gchar *filename);
```

Open *sound_resource* for reading and assign filename.

Parameters

sound_resource	the AgsSoundResource	
filename	the filename as string	

Returns

TRUE if operation was successful, otherwise FALSE.

Since: 3.0.0

ags_sound_resource_rw_open ()

```
gboolean
ags_sound_resource_rw_open (AgsSoundResource *sound_resource,
                            gchar *filename,
                            guint audio_channels,
                            guint samplerate,
                            gboolean create);
```

Open *sound_resource* for reading/writing and assign filename. Setting *create* to TRUE causes to create the file if it doesn't exist.

Parameters

sound_resource	the AgsSoundResource	
filename	the filename as string	
audio_channels	the audio channels count	
samplerate	the samplerate	
create	if TRUE create file, otherwise don't	

Returns

TRUE if operation was successful, otherwise FALSE.

Since: 3.0.0

ags_sound_resource_load ()

```
void
ags_sound_resource_load (AgsSoundResource *sound_resource);
```

Load audio data of *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	

Since: 3.0.0

ags_sound_resource_info ()

```
void
ags_sound_resource_info (AgsSoundResource *sound_resource,
                         guint *frame_count,
                         guint *loop_start,
                         guint *loop_end);
```

Get information about *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	
frame_count	return location of frame count	
loop_start	return location of loop-start	
loop_end	return location of loop-end	

Since: 3.0.0

ags_sound_resource_set_presets ()

```
void
ags_sound_resource_set_presets (AgsSoundResource *sound_resource,
                                guint channels,
                                guint samplerate,
                                guint buffer_size,
                                guint format);
```

Set presets of *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	
channels	channels to set	
samplerate	samplerate to set	
buffer_size	buffer-size to set	
format	format to set	

Since: 3.0.0

ags_sound_resource_get_presets ()

```
void
ags_sound_resource_get_presets (AgsSoundResource *sound_resource,
                                guint *channels,
                                guint *samplerate,
                                guint *buffer_size,
                                guint *format);
```

Get presets of *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	
channels	return location of channels	
samplerate	return location of samplerate	
buffer_size	return location of buffer-size	
format	return location of format	

Since: 3.0.0

ags_sound_resource_read ()

```
guint
ags_sound_resource_read (AgsSoundResource *sound_resource,
                         void *dbuffer,
                         guint daudio_channels,
                         guint audio_channel,
                         guint frame_count,
                         guint format);
```

Read *frame_count* number of frames from *sound_resource* and copy the data to *dbuffer* using *format* by skipping *daudio_channels*.

Parameters

sound_resource	the AgsSoundResource	
dbuffer	the destination buffer	
daudio_channels	destination buffer audio channel count	
audio_channel	the audio channel to read	
frame_count	the frame count to read	
format	the format to read	

Returns

the count of frames actually read

Since: 3.0.0

ags_sound_resource_write ()

```
void
ags_sound_resource_write (AgsSoundResource *sound_resource,
                          void *sbuffer,
                          guint saudio_channels,
                          guint audio_channel,
                          guint frame_count,
                          guint format);
```

Write *sbuffer* to *sound_resource* *frame_count* number of frames having *format* by skipping *saudio_channels*.

Parameters

sound_resource	the AgsSoundResource	
sbuffer	the source buffer	
saudio_channels	source buffer audio channel count	
audio_channel	the audio channel	
frame_count	the frame count to write	
format	the format to write	

Since: 3.0.0

ags_sound_resource_flush ()

```
void
ags_sound_resource_flush (AgsSoundResource *sound_resource);
```

Flush *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	

Since: 3.0.0

ags_sound_resource_seek ()

```
void
ags_sound_resource_seek (AgsSoundResource *sound_resource,
                         gint64 frame_count,
                         gint whence);
```

Seek the *sound_resource* *frame_count* from *whence* .

Parameters

sound_resource	the AgsSoundResource	
frame_count	the frame count	
whence	SEEK_SET, SEEK_CUR or SEEK_END	

Since: 3.0.0

ags_sound_resource_read_audio_signal ()

```
GList~*
ags_sound_resource_read_audio_signal (AgsSoundResource *sound_resource,
                                      GObject *soundcard,
                                      gint audio_channel);
```

Read audio signal from *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	
soundcard	the AgsSoundcard	
audio_channel	the audio channel or -1 for all	

Returns

a GList containing AgsAudioSignal.

[element-type AgsAudio.AudioSignal][transfer full]

Since: 3.0.0

ags_sound_resource_read_wave ()

```
GList~*
ags_sound_resource_read_wave (AgsSoundResource *sound_resource,
                              GObject *soundcard,
                              gint audio_channel,
                              guint64 x_offset,
                              gdouble delay,
                              guint attack);
```

Read wave from *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	
soundcard	the AgsSoundcard	
audio_channel	the audio channel or -1 for all	
x_offset	the x offset	
delay	the delay	
attack	the attack	

Returns

a GList containing AgsWave.

[element-type AgsAudio.Wave][transfer full]

Since: 3.0.0

ags_sound_resource_close ()

```
void
ags_sound_resource_close (AgsSoundResource *sound_resource);
```

Close *sound_resource* .

Parameters

sound_resource	the AgsSoundResource	

Since: 3.0.0

AGS_IS_SOUND_RESOURCE()

```
#define AGS_IS_SOUND_RESOURCE(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SOUND_RESOURCE))
```

AGS_IS_SOUND_RESOURCE_INTERFACE()

```
#define AGS_IS_SOUND_RESOURCE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SOUND_RESOURCE))
```

AGS_SOUND_RESOURCE()

```
#define AGS_SOUND_RESOURCE(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOUND_RESOURCE, AgsSoundResource))
```

AGS_SOUND_RESOURCE_GET_INTERFACE()

```
#define AGS_SOUND_RESOURCE_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SOUND_RESOURCE, AgsSoundResourceInterface))
```

AGS_SOUND_RESOURCE_INTERFACE()

```
#define AGS_SOUND_RESOURCE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SOUND_RESOURCE, AgsSoundResourceInterface))
```

ags_sound_resource_get_type ()

```
GType
ags_sound_resource_get_type ();
```

Types and Values

AGS_TYPE_SOUND_RESOURCE

```
#define AGS_TYPE_SOUND_RESOURCE                    (ags_sound_resource_get_type())
```

AgsSoundResource

```
typedef struct _AgsSoundResource AgsSoundResource;
```

struct AgsSoundResourceInterface

```
struct AgsSoundResourceInterface {
  GTypeInterface ginterface;

  gboolean (*open)(AgsSoundResource *sound_resource,
       gchar *filename);
  gboolean (*rw_open)(AgsSoundResource *sound_resource,
          gchar *filename,
          guint audio_channels, guint samplerate,
          gboolean create);

  void (*load)(AgsSoundResource *sound_resource);

  void (*info)(AgsSoundResource *sound_resource,
         guint *frame_count,
         guint *loop_start, guint *loop_end);

  void (*set_presets)(AgsSoundResource *sound_resource,
          guint channels,
```

```
          guint samplerate,
          guint buffer_size,
          guint format);
  void (*get_presets)(AgsSoundResource *sound_resource,
          guint *channels,
          guint *samplerate,
          guint *buffer_size,
          guint *format);

  /* read sample data */
  guint (*read)(AgsSoundResource *sound_resource,
    void *dbuffer, guint daudio_channels,
    guint audio_channel,
    guint frame_count, guint format);

  /* write sample data */
  void (*write)(AgsSoundResource *sound_resource,
    void *sbuffer, guint saudio_channels,
    guint audio_channel,
    guint frame_count, guint format);
  void (*flush)(AgsSoundResource *sound_resource);

  /* position */
  void (*seek)(AgsSoundResource *sound_resource,
         gint64 frame_count, gint whence);

  /* close */
  void (*close)(AgsSoundResource *sound_resource);
};
```

Chapter 19

Fx Engine - playback and recording

Multipurpose playback and recording or buffering of audio signals.

19.1 AgsFxPlaybackAudio

AgsFxPlaybackAudio — fx playback audio

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_fx_playback_audio_test_flags ()
void	ags_fx_playback_audio_set_flags ()
void	ags_fx_playback_audio_unset_flags ()
GList *	ags_fx_playback_audio_get_feed_audio_signal ()
void	ags_fx_playback_audio_add_feed_audio_signal ()
void	ags_fx_playback_audio_remove_feed_audio_signal ()
GList *	ags_fx_playback_audio_get_master_audio_signal ()
void	ags_fx_playback_audio_add_master_audio_signal ()
void	ags_fx_playback_audio_remove_master_audio_signal ()
void	ags_fx_playback_audio_open_audio_file ()
AgsFxPlaybackAudio *	ags_fx_playback_audio_new ()
#define	AGS_FX_PLAYBACK_AUDIO()
#define	AGS_FX_PLAYBACK_AUDIO_CLASS()
#define	AGS_FX_PLAYBACK_AUDIO_GET_CLASS()
#define	AGS_IS_FX_PLAYBACK_AUDIO()
#define	AGS_IS_FX_PLAYBACK_AUDIO_CLASS()
GType	ags_fx_playback_audio_get_type ()

Properties

AgsPort *	bpm	Read / Write
AgsPort *	capture-mode	Read / Write
AgsPort *	delay	Read / Write

AgsPort *	duration	Read / Write
AgsPort *	export	Read / Write
AgsPort *	file-audio-channels	Read / Write
AgsPort *	file-buffer-size	Read / Write
AgsPort *	file-format	Read / Write
AgsPort *	file-samplerate	Read / Write
AgsPort *	filename	Read / Write
AgsPort *	loop	Read / Write
AgsPort *	loop-end	Read / Write
AgsPort *	loop-start	Read / Write
AgsPort *	tact	Read / Write

Types and Values

#define	AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_START
#define	AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_END
enum	AgsFxPlaybackAudioFlags
enum	AgsFxPlaybackAudioCaptureMode
#define	AGS_TYPE_FX_PLAYBACK_AUDIO
struct	AgsFxPlaybackAudio
struct	AgsFxPlaybackAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxPlaybackAudio
```

Implemented Interfaces

AgsFxPlaybackAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_playback_audio.h>
```

Description

The AgsFxPlaybackAudio class provides ports to the effect processor.

Functions

ags_fx_playback_audio_test_flags ()

```
gboolean
ags_fx_playback_audio_test_flags (AgsFxPlaybackAudio *fx_playback_audio,
                                  guint flags);
```

Test *flags* of *fx_playback_audio*.

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
flags	the flags	

Returns

TRUE on success, otherwise FALSE

Since: 3.3.0

ags_fx_playback_audio_set_flags ()

```
void
ags_fx_playback_audio_set_flags (AgsFxPlaybackAudio *fx_playback_audio,
                                 guint flags);
```

Set *flags* of *fx_playback_audio* .

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
flags	the flags	

Since: 3.3.0

ags_fx_playback_audio_unset_flags ()

```
void
ags_fx_playback_audio_unset_flags (AgsFxPlaybackAudio *fx_playback_audio,
                                   guint flags);
```

Unset *flags* of *fx_playback_audio* .

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
flags	the flags	

Since: 3.3.0

ags_fx_playback_audio_get_feed_audio_signal ()

```
GList~*
ags_fx_playback_audio_get_feed_audio_signal
                                (AgsFxPlaybackAudio *fx_playback_audio);
```

Get feed audio_signal of *fx_playback_audio* .

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	

Returns

the GList containing feed audio_signal.

[element-type AgsAudio.AudioSignal][transfer full]

Since: 3.3.0

ags_fx_playback_audio_add_feed_audio_signal ()

```
void
ags_fx_playback_audio_add_feed_audio_signal
                                (AgsFxPlaybackAudio *fx_playback_audio,
                                 AgsAudioSignal *audio_signal);
```

Add *audio_signal* to *fx_playback_audio* .

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
audio_signal	the AgsAudioSignal	

Since: 3.3.0

ags_fx_playback_audio_remove_feed_audio_signal ()

```
void
ags_fx_playback_audio_remove_feed_audio_signal
                                (AgsFxPlaybackAudio *fx_playback_audio,
                                 AgsAudioSignal *audio_signal);
```

Remove *audio_signal* from *fx_playback_audio* .

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
audio_signal	the AgsAudioSignal	

Since: 3.3.0

ags_fx_playback_audio_get_master_audio_signal ()

```
GList~*
ags_fx_playback_audio_get_master_audio_signal
                                (AgsFxPlaybackAudio *fx_playback_audio);
```

Get master audio_signal of *fx_playback_audio* .

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	

Returns

the GList containing master audio_signal.

[element-type AgsAudio.AudioSignal][transfer full]

Since: 3.3.0

ags_fx_playback_audio_add_master_audio_signal ()

```
void
ags_fx_playback_audio_add_master_audio_signal
                                (AgsFxPlaybackAudio *fx_playback_audio,
                                 AgsAudioSignal *audio_signal);
```

Add `audio_signal` to `fx_playback_audio`.

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
audio_signal	the AgsAudioSignal	

Since: 3.3.0

ags_fx_playback_audio_remove_master_audio_signal ()

```
void
ags_fx_playback_audio_remove_master_audio_signal
                                (AgsFxPlaybackAudio *fx_playback_audio,
                                 AgsAudioSignal *audio_signal);
```

Remove `audio_signal` from `fx_playback_audio`.

Parameters

fx_playback_audio	the AgsFxPlaybackAudio	
audio_signal	the AgsAudioSignal	

Since: 3.3.0

ags_fx_playback_audio_open_audio_file ()

```
void
ags_fx_playback_audio_open_audio_file (AgsFxPlaybackAudio *fx_playback_audio);
```

Open audio file of `fx_playback_audio`.

Parameters

fx_playback_audio	the AgsFxPlaybackAudio

Since: 3.3.0

ags_fx_playback_audio_new ()

```
AgsFxPlaybackAudio~*
ags_fx_playback_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxPlaybackAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxPlaybackAudio

Since: 3.3.0

AGS_FX_PLAYBACK_AUDIO()

```
#define AGS_FX_PLAYBACK_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_AUDIO, AgsFxPlaybackAudio))
```

AGS_FX_PLAYBACK_AUDIO_CLASS()

```
#define AGS_FX_PLAYBACK_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PLAYBACK_AUDIO, AgsFxPlaybackAudioClass))
```

AGS_FX_PLAYBACK_AUDIO_GET_CLASS()

```
#define AGS_FX_PLAYBACK_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_AUDIO, AgsFxPlaybackAudioClass))
```

AGS_IS_FX_PLAYBACK_AUDIO()

```
#define AGS_IS_FX_PLAYBACK_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_AUDIO))
```

AGS_IS_FX_PLAYBACK_AUDIO_CLASS()

```
#define AGS_IS_FX_PLAYBACK_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PLAYBACK_AUDIO))
```

ags_fx_playback_audio_get_type ()

```
GType
ags_fx_playback_audio_get_type ();
```

Types and Values

AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_START

```
#define AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_START (0)
```

AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_END

```
#define AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_END (64)
```

enum AgsFxPlaybackAudioFlags

Enum values to enable specific feature of AgsFxPlaybackAudio.

Members

AGS_FX_PLAYBACK_AUDIO_PLAY	do play sound
AGS_FX_PLAYBACK_AUDIO_RECORD	do record sound
AGS_FX_PLAYBACK_AUDIO_FEED	do feed au- dio sig- nal
AGS_FX_PLAYBACK_AUDIO_MASTER	do mas- ter au- dio sig- nal

enum AgsFxPlaybackAudioCaptureMode

Enum values to specify capture mode of AgsFxPlaybackAudio.

Members

AGS_FX_PLAYBACK_AUDIO_CAPTURE_MODE_DISCARD	discard capture
AGS_FX_PLAYBACK_AUDIO_CAPTURE_MODE_NONE	capture none and preserve data
AGS_FX_PLAYBACK_AUDIO_CAPTURE_MODE_REPLACE	capture replace any sound
AGS_FX_PLAYBACK_AUDIO_CAPTURE_MODE_MIX	capture mix with existing sound

AGS_TYPE_FX_PLAYBACK_AUDIO

```
#define AGS_TYPE_FX_PLAYBACK_AUDIO                (ags_fx_playback_audio_get_type())
```

struct AgsFxPlaybackAudio

```
struct AgsFxPlaybackAudio;
```

struct AgsFxPlaybackAudioClass

```
struct AgsFxPlaybackAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

Property Details

The "bpm" property

```
  "bpm"                      AgsPort~*
```

The beats per minute.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "capture-mode" property

```
"capture-mode"              AgsPort~*
```

The capture mode.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "delay" property

```
"delay"                     AgsPort~*
```

The playback's delay.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "duration" property

```
"duration"                  AgsPort~*
```

The playback's duration.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "export" property

```
"export"                    AgsPort~*
```

The export switch.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "file-audio-channels" property

```
"file-audio-channels"       AgsPort~*
```

The file's audio channel count.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "file-buffer-size" property

```
"file-buffer-size"          AgsPort~*
```

The file's buffer size.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "file-format" property

```
"file-format"               AgsPort~*
```

The file's format.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "file-samplerate" property

```
"file-samplerate"           AgsPort~*
```

The file's samplerate.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "filename" property

```
"filename"                  AgsPort~*
```

The filename to export.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "loop" property

```
"loop"                      AgsPort~*
```

Do loop.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "loop-end" property

```
"loop-end"                  AgsPort~*
```

The playback's loop-end.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "loop-start" property

```
"loop-start"                AgsPort~*
```

The playback's loop-start.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

The "tact" property

```
"tact"                      AgsPort~*
```

The tact segmentation.

Owner: AgsFxPlaybackAudio

Flags: Read / Write

Since: 3.3.0

19.2 AgsFxPlaybackAudioProcessor

AgsFxPlaybackAudioProcessor — fx playback audio processor

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_fx_playback_audio_processor_data_put ()
void	ags_fx_playback_audio_processor_data_get ()
void	ags_fx_playback_audio_processor_play ()
void	ags_fx_playback_audio_processor_record ()
void	ags_fx_playback_audio_processor_feed ()
void	ags_fx_playback_audio_processor_master ()
void	ags_fx_playback_audio_processor_counter_change ()
AgsFxPlaybackAudioProcessor *	ags_fx_playback_audio_processor_new ()
#define	AGS_FX_PLAYBACK_AUDIO_PROCESSOR()

#define	AGS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_PLAYBACK_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR()
#define	AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_playback_audio_processor_get_type ()

Types and Values

enum	AgsFxPlaybackAudioProcessorDataMode
#define	AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR
struct	AgsFxPlaybackAudioProcessor
struct	AgsFxPlaybackAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxPlaybackAudioProcessor
```

Implemented Interfaces

AgsFxPlaybackAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_playback_audio_processor.h>
```

Description

The AgsFxPlaybackAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_playback_audio_processor_data_put ()

```
void
ags_fx_playback_audio_processor_data_put
                                (AgsFxPlaybackAudioProcessor *fx_playback_audio_processor,
                                 AgsBuffer *buffer,
                                 guint data_mode);
```

ags_fx_playback_audio_processor_data_get ()

```
void
ags_fx_playback_audio_processor_data_get
                                (AgsFxPlaybackAudioProcessor *fx_playback_audio_processor,
                                 AgsBuffer *buffer,
                                 guint data_mode);
```

ags_fx_playback_audio_processor_play ()

```
void
ags_fx_playback_audio_processor_play (AgsFxPlaybackAudioProcessor * ↩
    fx_playback_audio_processor);
```

ags_fx_playback_audio_processor_record ()

```
void
ags_fx_playback_audio_processor_record
                                (AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
```

ags_fx_playback_audio_processor_feed ()

```
void
ags_fx_playback_audio_processor_feed (AgsFxPlaybackAudioProcessor * ↩
    fx_playback_audio_processor);
```

ags_fx_playback_audio_processor_master ()

```
void
ags_fx_playback_audio_processor_master
                                (AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
```

ags_fx_playback_audio_processor_counter_change ()

```
void
ags_fx_playback_audio_processor_counter_change
                                (AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
```

ags_fx_playback_audio_processor_new ()

```
AgsFxPlaybackAudioProcessor~*
ags_fx_playback_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxPlaybackAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxPlaybackAudioProcessor

Since: 3.3.0

AGS_FX_PLAYBACK_AUDIO_PROCESSOR()

```
#define AGS_FX_PLAYBACK_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR, AgsFxPlaybackAudioProcessor))
```

AGS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR, AgsFxPlaybackAudioProcessor))
```

AGS_FX_PLAYBACK_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_PLAYBACK_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR, AgsFxPlaybackAudioProcessorClass))
```

AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR))
```

AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR))
```

ags_fx_playback_audio_processor_get_type ()

```
GType
ags_fx_playback_audio_processor_get_type
                                ();
```

Types and Values

enum AgsFxPlaybackAudioProcessorDataMode

Enum values to enable specific data mode of AgsFxPlaybackAudioProcessor.

Members

AGS_FX_PLAYBACK_AUDIO_PROCESSOR_DATA_MODE_PLAY	data mode play sound
AGS_FX_PLAYBACK_AUDIO_PROCESSOR_DATA_MODE_RECORD	data mode record sound

AGS_FX_PLAYBACK_AUDIO_PROCESSOR_DATA_MODE_FEED	data mode feed audio signal
AGS_FX_PLAYBACK_AUDIO_PROCESSOR_DATA_MODE_MASTER	data mode master audio signal

AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR                ( ↩
    ags_fx_playback_audio_processor_get_type())
```

struct AgsFxPlaybackAudioProcessor

```
struct AgsFxPlaybackAudioProcessor;
```

struct AgsFxPlaybackAudioProcessorClass

```
struct AgsFxPlaybackAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;

  void (*data_put)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor,
       AgsBuffer *buffer,
       guint data_mode);
  void (*data_get)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor,
       AgsBuffer *buffer,
       guint data_mode);

  void (*play)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
  void (*record)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
  void (*feed)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
  void (*master)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);

  void (*counter_change)(AgsFxPlaybackAudioProcessor *fx_playback_audio_processor);
};
```

19.3 AgsFxPlaybackAudioSignal

AgsFxPlaybackAudioSignal — fx playback audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPlaybackAudioSignal *	ags_fx_playback_audio_signal_new ()
#define	AGS_FX_PLAYBACK_AUDIO_SIGNAL()
#define	AGS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_PLAYBACK_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL()
#define	AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS()
GType	ags_fx_playback_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL
struct	AgsFxPlaybackAudioSignal
struct	AgsFxPlaybackAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxPlaybackAudioSignal
```

Implemented Interfaces

AgsFxPlaybackAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_playback_audio_signal.h>
```

Description

The AgsFxPlaybackAudioSignal class provides ports to the effect processor.

Functions

ags_fx_playback_audio_signal_new ()

```
AgsFxPlaybackAudioSignal~*
ags_fx_playback_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxPlaybackAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxPlaybackAudioSignal

Since: 3.3.0

AGS_FX_PLAYBACK_AUDIO_SIGNAL()

```
#define AGS_FX_PLAYBACK_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL, AgsFxPlaybackAudioSignal))
```

AGS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL, AgsFxPlaybackAudioSignalClass))
```

AGS_FX_PLAYBACK_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_PLAYBACK_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL, AgsFxPlaybackAudioSignalClass))
```

AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL()

```
#define AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL))
```

AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL))
```

ags_fx_playback_audio_signal_get_type ()

```
GType
ags_fx_playback_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL                ( ↩
    ags_fx_playback_audio_signal_get_type())
```

struct AgsFxPlaybackAudioSignal

```
struct AgsFxPlaybackAudioSignal;
```

struct AgsFxPlaybackAudioSignalClass

```
struct AgsFxPlaybackAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

19.4 AgsFxPlaybackChannel

AgsFxPlaybackChannel — fx playback channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPlaybackChannel *	ags_fx_playback_channel_new ()
#define	AGS_FX_PLAYBACK_CHANNEL()
#define	AGS_FX_PLAYBACK_CHANNEL_CLASS()
#define	AGS_FX_PLAYBACK_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_PLAYBACK_CHANNEL()
#define	AGS_IS_FX_PLAYBACK_CHANNEL_CLASS()
GType	ags_fx_playback_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_PLAYBACK_CHANNEL
struct	AgsFxPlaybackChannel
struct	AgsFxPlaybackChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxPlaybackChannel
```

Implemented Interfaces

AgsFxPlaybackChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_playback_channel.h>
```

Description

The AgsFxPlaybackChannel class provides ports to the effect processor.

Functions

ags_fx_playback_channel_new ()

```
AgsFxPlaybackChannel~*
ags_fx_playback_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxPlaybackChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxPlaybackChannel

Since: 3.3.0

AGS_FX_PLAYBACK_CHANNEL()

```
#define AGS_FX_PLAYBACK_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_CHANNEL, AgsFxPlaybackChannel))
```

AGS_FX_PLAYBACK_CHANNEL_CLASS()

```
#define AGS_FX_PLAYBACK_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PLAYBACK_CHANNEL, AgsFxPlaybackChannel))
```

AGS_FX_PLAYBACK_CHANNEL_GET_CLASS()

```
#define AGS_FX_PLAYBACK_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_CHANNEL, AgsFxPlaybackChannelClass))
```

AGS_IS_FX_PLAYBACK_CHANNEL()

```
#define AGS_IS_FX_PLAYBACK_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_CHANNEL))
```

AGS_IS_FX_PLAYBACK_CHANNEL_CLASS()

```
#define AGS_IS_FX_PLAYBACK_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PLAYBACK_CHANNEL))
```

ags_fx_playback_channel_get_type ()

```
GType
ags_fx_playback_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_PLAYBACK_CHANNEL

```
#define AGS_TYPE_FX_PLAYBACK_CHANNEL                (ags_fx_playback_channel_get_type())
```

struct AgsFxPlaybackChannel

```
struct AgsFxPlaybackChannel;
```

struct AgsFxPlaybackChannelClass

```
struct AgsFxPlaybackChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

19.5 AgsFxPlaybackChannelProcessor

AgsFxPlaybackChannelProcessor — fx playback channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPlaybackChannelProcessor *	ags_fx_playback_channel_processor_new ()
#define	AGS_FX_PLAYBACK_CHANNEL_PROCESSOR()
#define	AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_playback_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR
struct	AgsFxPlaybackChannelProcessor
struct	AgsFxPlaybackChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxPlaybackChannelProcessor
```

Implemented Interfaces

AgsFxPlaybackChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_playback_channel_processor.h>
```

Description

The AgsFxPlaybackChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_playback_channel_processor_new ()

```
AgsFxPlaybackChannelProcessor~*
ags_fx_playback_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxPlaybackChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxPlaybackChannelProcessor

Since: 3.3.0

AGS_FX_PLAYBACK_CHANNEL_PROCESSOR()

```
#define AGS_FX_PLAYBACK_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR, AgsFxPlaybackChannelProcessor))
```

AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR, AgsFxPlaybackChannelProcessorClass))
```

AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR, AgsFxPlaybackChannelProcessorClass))
```

AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR))
```

AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR))
```

ags_fx_playback_channel_processor_get_type ()

```
GType
ags_fx_playback_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR                 ( ↩
    ags_fx_playback_channel_processor_get_type())
```

struct AgsFxPlaybackChannelProcessor

```
struct AgsFxPlaybackChannelProcessor;
```

struct AgsFxPlaybackChannelProcessorClass

```
struct AgsFxPlaybackChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

19.6 AgsFxPlaybackRecycling

AgsFxPlaybackRecycling — fx playback recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPlaybackRecycling *	ags_fx_playback_recycling_new ()
#define	AGS_FX_PLAYBACK_RECYCLING()
#define	AGS_FX_PLAYBACK_RECYCLING_CLASS()
#define	AGS_FX_PLAYBACK_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_PLAYBACK_RECYCLING()
#define	AGS_IS_FX_PLAYBACK_RECYCLING_CLASS()
GType	ags_fx_playback_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_PLAYBACK_RECYCLING
struct	AgsFxPlaybackRecycling
struct	AgsFxPlaybackRecyclingClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallRecycling
        &#x2570;&#x2500;&#x2500; AgsFxPlaybackRecycling
```

Implemented Interfaces

AgsFxPlaybackRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_playback_recycling.h>
```

Description

The AgsFxPlaybackRecycling class provides ports to the effect processor.

Functions

ags_fx_playback_recycling_new ()

```
AgsFxPlaybackRecycling~*
ags_fx_playback_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxPlaybackRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxPlaybackRecycling

Since: 3.3.0

AGS_FX_PLAYBACK_RECYCLING()

```
#define AGS_FX_PLAYBACK_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_RECYCLING, AgsFxPlaybackRecycling))
```

AGS_FX_PLAYBACK_RECYCLING_CLASS()

```
#define AGS_FX_PLAYBACK_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PLAYBACK_RECYCLING, AgsFxPlaybackRecyclingClass))
```

AGS_FX_PLAYBACK_RECYCLING_GET_CLASS()

```
#define AGS_FX_PLAYBACK_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_RECYCLING, AgsFxPlaybackRecyclingClass))
```

AGS_IS_FX_PLAYBACK_RECYCLING()

```
#define AGS_IS_FX_PLAYBACK_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PLAYBACK_RECYCLING))
```

AGS_IS_FX_PLAYBACK_RECYCLING_CLASS()

```
#define AGS_IS_FX_PLAYBACK_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PLAYBACK_RECYCLING))
```

ags_fx_playback_recycling_get_type ()

```
GType
ags_fx_playback_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_PLAYBACK_RECYCLING

```
#define AGS_TYPE_FX_PLAYBACK_RECYCLING                (ags_fx_playback_recycling_get_type() ↩
    )
```

struct AgsFxPlaybackRecycling

```
struct AgsFxPlaybackRecycling;
```

struct AgsFxPlaybackRecyclingClass

```
struct AgsFxPlaybackRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

19.7 AgsFxBufferAudio

AgsFxBufferAudio — fx buffer audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxBufferAudio *	ags_fx_buffer_audio_new ()
#define	AGS_FX_BUFFER_AUDIO()
#define	AGS_FX_BUFFER_AUDIO_CLASS()
#define	AGS_FX_BUFFER_AUDIO_GET_CLASS()
#define	AGS_IS_FX_BUFFER_AUDIO()
#define	AGS_IS_FX_BUFFER_AUDIO_CLASS()
GType	ags_fx_buffer_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_BUFFER_AUDIO
struct	AgsFxBufferAudio
struct	AgsFxBufferAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxBufferAudio
```

Implemented Interfaces

AgsFxBufferAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_buffer_audio.h>
```

Description

The AgsFxBufferAudio class provides ports to the effect processor.

Functions

ags_fx_buffer_audio_new ()

```
AgsFxBufferAudio~*
ags_fx_buffer_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxBufferAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxBufferAudio

Since: 3.3.0

AGS_FX_BUFFER_AUDIO()

```
#define AGS_FX_BUFFER_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO, AgsFxBufferAudio))
```

AGS_FX_BUFFER_AUDIO_CLASS()

```
#define AGS_FX_BUFFER_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO, AgsFxBufferAudioClass))
```

AGS_FX_BUFFER_AUDIO_GET_CLASS()

```
#define AGS_FX_BUFFER_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO, AgsFxBufferAudioClass))
```

AGS_IS_FX_BUFFER_AUDIO()

```
#define AGS_IS_FX_BUFFER_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO))
```

AGS_IS_FX_BUFFER_AUDIO_CLASS()

```
#define AGS_IS_FX_BUFFER_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO))
```

ags_fx_buffer_audio_get_type ()

```
GType
ags_fx_buffer_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_BUFFER_AUDIO

```
#define AGS_TYPE_FX_BUFFER_AUDIO                (ags_fx_buffer_audio_get_type())
```

struct AgsFxBufferAudio

```
struct AgsFxBufferAudio;
```

struct AgsFxBufferAudioClass

```
struct AgsFxBufferAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

19.8 AgsFxBufferAudioProcessor

AgsFxBufferAudioProcessor — fx buffer audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxBufferAudioProcessor *	ags_fx_buffer_audio_processor_new ()
#define	AGS_FX_BUFFER_AUDIO_PROCESSOR()
#define	AGS_FX_BUFFER_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_BUFFER_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_BUFFER_AUDIO_PROCESSOR()
#define	AGS_IS_FX_BUFFER_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_buffer_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR
struct	AgsFxBufferAudioProcessor
struct	AgsFxBufferAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxBufferAudioProcessor
```

Implemented Interfaces

AgsFxBufferAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_buffer_audio_processor.h>
```

Description

The AgsFxBufferAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_buffer_audio_processor_new ()

```
AgsFxBufferAudioProcessor~*
ags_fx_buffer_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxBufferAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxBufferAudioProcessor

Since: 3.3.0

AGS_FX_BUFFER_AUDIO_PROCESSOR()

```
#define AGS_FX_BUFFER_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR, AgsFxBufferAudioProcessor))
```

AGS_FX_BUFFER_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_BUFFER_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR, AgsFxBufferAudioProcessorClass))
```

AGS_FX_BUFFER_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_BUFFER_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR, AgsFxBufferAudioProcessorClass))
```

AGS_IS_FX_BUFFER_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_BUFFER_AUDIO_PROCESSOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR))
```

AGS_IS_FX_BUFFER_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_BUFFER_AUDIO_PROCESSOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR))
```

ags_fx_buffer_audio_processor_get_type ()

```
GType
ags_fx_buffer_audio_processor_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR                 ( ↩
    ags_fx_buffer_audio_processor_get_type())
```

struct AgsFxBufferAudioProcessor

```
struct AgsFxBufferAudioProcessor;
```

struct AgsFxBufferAudioProcessorClass

```
struct AgsFxBufferAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

19.9 AgsFxBufferAudioSignal

AgsFxBufferAudioSignal — fx buffer audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxBufferAudioSignal *	ags_fx_buffer_audio_signal_new ()
#define	AGS_FX_BUFFER_AUDIO_SIGNAL()
#define	AGS_FX_BUFFER_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_BUFFER_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_BUFFER_AUDIO_SIGNAL()
#define	AGS_IS_FX_BUFFER_AUDIO_SIGNAL_CLASS()
GType	ags_fx_buffer_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL
struct	AgsFxBufferAudioSignal
struct	AgsFxBufferAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxBufferAudioSignal
```

Implemented Interfaces

AgsFxBufferAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_buffer_audio_signal.h>
```

Description

The AgsFxBufferAudioSignal class provides ports to the effect processor.

Functions

ags_fx_buffer_audio_signal_new ()

```
AgsFxBufferAudioSignal~*
ags_fx_buffer_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxBufferAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxBufferAudioSignal

Since: 3.3.0

AGS_FX_BUFFER_AUDIO_SIGNAL()

```
#define AGS_FX_BUFFER_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL, AgsFxBufferAudioSignal))
```

AGS_FX_BUFFER_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_BUFFER_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL, AgsFxBufferAudioSignalClass))
```

AGS_FX_BUFFER_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_BUFFER_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL, AgsFxBufferAudioSignalClass))
```

AGS_IS_FX_BUFFER_AUDIO_SIGNAL()

```
#define AGS_IS_FX_BUFFER_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL))
```

AGS_IS_FX_BUFFER_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_BUFFER_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL))
```

ags_fx_buffer_audio_signal_get_type ()

```
GType
ags_fx_buffer_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL                (ags_fx_buffer_audio_signal_get_type ↩
    ())
```

struct AgsFxBufferAudioSignal

```
struct AgsFxBufferAudioSignal;
```

struct AgsFxBufferAudioSignalClass

```
struct AgsFxBufferAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

19.10 AgsFxBufferChannel

AgsFxBufferChannel — fx buffer channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_BUFFER_CHANNEL_INPUT_DATA()
#define	AGS_FX_BUFFER_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxBufferChannelInputData *	ags_fx_buffer_channel_input_data_alloc ()
void	ags_fx_buffer_channel_input_data_free ()
GRecMutex *	ags_fx_buffer_channel_input_data_get_strct_mutex ()
gpointer	ags_fx_buffer_channel_input_get_parent ()
gpointer	ags_fx_buffer_channel_input_get_destination ()
AgsFxBufferChannelInputData *	ags_fx_buffer_channel_get_input_data ()
AgsFxBufferChannel *	ags_fx_buffer_channel_new ()
#define	AGS_FX_BUFFER_CHANNEL()
#define	AGS_FX_BUFFER_CHANNEL_CLASS()
#define	AGS_FX_BUFFER_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_BUFFER_CHANNEL()
#define	AGS_IS_FX_BUFFER_CHANNEL_CLASS()
GType	ags_fx_buffer_channel_get_type ()

Types and Values

struct	AgsFxBufferChannelInputData
#define	AGS_TYPE_FX_BUFFER_CHANNEL
struct	AgsFxBufferChannel
struct	AgsFxBufferChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxBufferChannel
```

Implemented Interfaces

AgsFxBufferChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_buffer_channel.h>
```

Description

The AgsFxBufferChannel class provides ports to the effect processor.

Functions

AGS_FX_BUFFER_CHANNEL_INPUT_DATA()

```
#define AGS_FX_BUFFER_CHANNEL_INPUT_DATA(ptr) ((AgsFxBufferChannelInputData *) (ptr))
```

AGS_FX_BUFFER_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_BUFFER_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&((( ↩
    AgsFxBufferChannelInputData *)(ptr))->strct_mutex))
```

ags_fx_buffer_channel_input_data_alloc ()

```
AgsFxBufferChannelInputData~*
ags_fx_buffer_channel_input_data_alloc
                                ();
```

Allocate AgsFxBufferChannelInputData

Returns

the new AgsFxBufferChannelInputData.

[type gpointer][transfer full]

Since: 3.3.0

ags_fx_buffer_channel_input_data_free ()

```
void
ags_fx_buffer_channel_input_data_free (AgsFxBufferChannelInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxBufferChannelInputData.	*[type gpointer][transfer full]*

Since: 3.3.0

ags_fx_buffer_channel_input_data_get_strct_mutex ()

```
GRecMutex~*
ags_fx_buffer_channel_input_data_get_strct_mutex
                                (AgsFxBufferChannelInputData *input_data);
```

Get structure mutex.

Parameters

input_data	the AgsFxBufferChannelInputData.	*[type gpointer][transfer none]*

Returns

the GRecMutex to lock *input_data* .

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_buffer_channel_input_get_parent ()

```
gpointer
ags_fx_buffer_channel_input_get_parent
                                (AgsFxBufferChannelInputData *input_data);
```

Get parent of *input_data* .

Parameters

input_data	the AgsFxBufferChannelInputData.	*[type gpointer][transfer none]*

Returns

the parent.

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_buffer_channel_input_get_destination ()

```
gpointer
ags_fx_buffer_channel_input_get_destination
                                (AgsFxBufferChannelInputData *input_data);
```

Get destination of *input_data* .

Parameters

input_data	the AgsFxBufferChannelInputData.	*[type gpointer][transfer none]*

Returns

the destination.

[element-type AgsAudio.Recycling AgsAudio.AudioSignal][transfer none]

Since: 3.3.0

ags_fx_buffer_channel_get_input_data ()

```
AgsFxBufferChannelInputData~*
ags_fx_buffer_channel_get_input_data (AgsFxBufferChannel *fx_buffer_channel,
                                      gint sound_scope);
```

Get input data from `fx_buffer_channel` by `sound_scope` .

Parameters

fx_buffer_channel	the AgsFxBufferChannel	
sound_scope	the sound scope	

Returns

the matching AgsFxBufferChannelInputData.

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_buffer_channel_new ()

```
AgsFxBufferChannel~*
ags_fx_buffer_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxBufferChannel

Parameters

channel	the AgsChannel	

Returns

the new AgsFxBufferChannel

Since: 3.3.0

AGS_FX_BUFFER_CHANNEL()

```
#define AGS_FX_BUFFER_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_BUFFER_CHANNEL, AgsFxBufferChannel))
```

AGS_FX_BUFFER_CHANNEL_CLASS()

```
#define AGS_FX_BUFFER_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_BUFFER_CHANNEL, AgsFxBufferChannelClass))
```

AGS_FX_BUFFER_CHANNEL_GET_CLASS()

```
#define AGS_FX_BUFFER_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_BUFFER_CHANNEL, AgsFxBufferChannelClass))
```

AGS_IS_FX_BUFFER_CHANNEL()

```
#define AGS_IS_FX_BUFFER_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_BUFFER_CHANNEL))
```

AGS_IS_FX_BUFFER_CHANNEL_CLASS()

```
#define AGS_IS_FX_BUFFER_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_BUFFER_CHANNEL))
```

ags_fx_buffer_channel_get_type ()

```
GType
ags_fx_buffer_channel_get_type ();
```

Types and Values

struct AgsFxBufferChannelInputData

```
struct AgsFxBufferChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  GHashTable *destination;
  GHashTable *resample_cache;
};
```

AGS_TYPE_FX_BUFFER_CHANNEL

```
#define AGS_TYPE_FX_BUFFER_CHANNEL                (ags_fx_buffer_channel_get_type())
```

struct AgsFxBufferChannel

```
struct AgsFxBufferChannel;
```

struct AgsFxBufferChannelClass

```
struct AgsFxBufferChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

19.11 AgsFxBufferChannelProcessor

AgsFxBufferChannelProcessor — fx buffer channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxBufferChannelProcessor *	ags_fx_buffer_channel_processor_new ()
#define	AGS_FX_BUFFER_CHANNEL_PROCESSOR()
#define	AGS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_BUFFER_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_buffer_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR
struct	AgsFxBufferChannelProcessor
struct	AgsFxBufferChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxBufferChannelProcessor
```

Implemented Interfaces

AgsFxBufferChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_buffer_channel_processor.h>
```

Description

The AgsFxBufferChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_buffer_channel_processor_new ()

```
AgsFxBufferChannelProcessor~*
ags_fx_buffer_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxBufferChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxBufferChannelProcessor

Since: 3.3.0

AGS_FX_BUFFER_CHANNEL_PROCESSOR()

```
#define AGS_FX_BUFFER_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR, AgsFxBufferChannelProcessor))
```

AGS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR, AgsFxBufferChannelProcessorClass))
```

AGS_FX_BUFFER_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_BUFFER_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR, AgsFxBufferChannelProcessorClass))
```

AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR))
```

AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR))
```

ags_fx_buffer_channel_processor_get_type ()

```
GType
ags_fx_buffer_channel_processor_get_type
                                    ();
```

Types and Values

AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR                ( ↩
    ags_fx_buffer_channel_processor_get_type())
```

struct AgsFxBufferChannelProcessor

```
struct AgsFxBufferChannelProcessor;
```

struct AgsFxBufferChannelProcessorClass

```
struct AgsFxBufferChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

19.12 AgsFxBufferRecycling

AgsFxBufferRecycling — fx buffer recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxBufferRecycling *	ags_fx_buffer_recycling_new ()
#define	AGS_FX_BUFFER_RECYCLING()
#define	AGS_FX_BUFFER_RECYCLING_CLASS()
#define	AGS_FX_BUFFER_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_BUFFER_RECYCLING()
#define	AGS_IS_FX_BUFFER_RECYCLING_CLASS()
GType	ags_fx_buffer_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_BUFFER_RECYCLING
struct	AgsFxBufferRecycling
struct	AgsFxBufferRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxBufferRecycling
```

Implemented Interfaces

AgsFxBufferRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_buffer_recycling.h>
```

Description

The AgsFxBufferRecycling class provides ports to the effect processor.

Functions

ags_fx_buffer_recycling_new ()

```
AgsFxBufferRecycling~*
ags_fx_buffer_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxBufferRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxBufferRecycling

Since: 3.3.0

AGS_FX_BUFFER_RECYCLING()

```
#define AGS_FX_BUFFER_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_BUFFER_RECYCLING, AgsFxBufferRecycling))
```

AGS_FX_BUFFER_RECYCLING_CLASS()

```
#define AGS_FX_BUFFER_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_BUFFER_RECYCLING, AgsFxBufferRecyclingClass))
```

AGS_FX_BUFFER_RECYCLING_GET_CLASS()

```
#define AGS_FX_BUFFER_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_BUFFER_RECYCLING, AgsFxBufferRecyclingClass))
```

AGS_IS_FX_BUFFER_RECYCLING()

```
#define AGS_IS_FX_BUFFER_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_BUFFER_RECYCLING))
```

AGS_IS_FX_BUFFER_RECYCLING_CLASS()

```
#define AGS_IS_FX_BUFFER_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_BUFFER_RECYCLING))
```

ags_fx_buffer_recycling_get_type ()

```
GType
ags_fx_buffer_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_BUFFER_RECYCLING

```
#define AGS_TYPE_FX_BUFFER_RECYCLING                (ags_fx_buffer_recycling_get_type())
```

struct AgsFxBufferRecycling

```
struct AgsFxBufferRecycling;
```

struct AgsFxBufferRecyclingClass

```
struct AgsFxBufferRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

Chapter 20

Fx Engine - pattern and notation

Pattern and notation based multipurpose playback, capture or feeding notes to audio signals.

20.1 AgsFxPatternAudio

AgsFxPatternAudio — fx pattern audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPatternAudio *	ags_fx_pattern_audio_new ()
#define	AGS_FX_PATTERN_AUDIO()
#define	AGS_FX_PATTERN_AUDIO_CLASS()
#define	AGS_FX_PATTERN_AUDIO_GET_CLASS()
#define	AGS_IS_FX_PATTERN_AUDIO()
#define	AGS_IS_FX_PATTERN_AUDIO_CLASS()
GType	ags_fx_pattern_audio_get_type ()

Properties

AgsPort *	bank-index-0	Read / Write
AgsPort *	bank-index-1	Read / Write
AgsPort *	bpm	Read / Write
AgsPort *	delay	Read / Write
AgsPort *	duration	Read / Write
AgsPort *	loop	Read / Write
AgsPort *	loop-end	Read / Write
AgsPort *	loop-start	Read / Write
AgsPort *	tact	Read / Write

Types and Values

#define	AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_START
#define	AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_END
#define	AGS_TYPE_FX_PATTERN_AUDIO
struct	AgsFxPatternAudio
struct	AgsFxPatternAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxPatternAudio
```

Implemented Interfaces

AgsFxPatternAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_pattern_audio.h>
```

Description

The AgsFxPatternAudio class provides ports to the effect processor.

Functions

ags_fx_pattern_audio_new ()

```
AgsFxPatternAudio~*
ags_fx_pattern_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxPatternAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxPatternAudio

Since: 3.3.0

AGS_FX_PATTERN_AUDIO()

```
#define AGS_FX_PATTERN_AUDIO(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO, AgsFxPatternAudio))
```

AGS_FX_PATTERN_AUDIO_CLASS()

```
#define AGS_FX_PATTERN_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO, AgsFxPatternAudioClass))
```

AGS_FX_PATTERN_AUDIO_GET_CLASS()

```
#define AGS_FX_PATTERN_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO, AgsFxPatternAudioClass))
```

AGS_IS_FX_PATTERN_AUDIO()

```
#define AGS_IS_FX_PATTERN_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO))
```

AGS_IS_FX_PATTERN_AUDIO_CLASS()

```
#define AGS_IS_FX_PATTERN_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO))
```

ags_fx_pattern_audio_get_type ()

```
GType
ags_fx_pattern_audio_get_type ();
```

Types and Values

AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_START

```
#define AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_START (0)
```

AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_END

```
#define AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_END (16)
```

AGS_TYPE_FX_PATTERN_AUDIO

```
#define AGS_TYPE_FX_PATTERN_AUDIO                (ags_fx_pattern_audio_get_type())
```

struct AgsFxPatternAudio

```
struct AgsFxPatternAudio;
```

struct AgsFxPatternAudioClass

```
struct AgsFxPatternAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

Property Details

The "bank-index-0" property

```
  "bank-index-0"              AgsPort~*
```

Selected bank index 0.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "bank-index-1" property

```
  "bank-index-1"              AgsPort~*
```

Selected bank index 1.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "bpm" property

```
  "bpm"                       AgsPort~*
```

The beats per minute.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "delay" property

```
  "delay"                     AgsPort~*
```

The pattern's delay.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "duration" property

```
"duration"                  AgsPort~*
```

The pattern's duration.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "loop" property

```
"loop"                      AgsPort~*
```

Do loop.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "loop-end" property

```
"loop-end"                  AgsPort~*
```

The pattern's loop-end.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "loop-start" property

```
"loop-start"                AgsPort~*
```

The pattern's loop-start.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

The "tact" property

```
"tact"                      AgsPort~*
```

The tact segmentation.

Owner: AgsFxPatternAudio

Flags: Read / Write

Since: 3.3.0

20.2 AgsFxPatternAudioProcessor

AgsFxPatternAudioProcessor — fx pattern audio processor

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_fx_pattern_audio_processor_key_on ()
void	ags_fx_pattern_audio_processor_play ()
void	ags_fx_pattern_audio_processor_counter_change ()
AgsFxPatternAudioProcessor *	ags_fx_pattern_audio_processor_new ()
#define	AGS_FX_PATTERN_AUDIO_PROCESSOR()
#define	AGS_FX_PATTERN_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_PATTERN_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_PATTERN_AUDIO_PROCESSOR()
#define	AGS_IS_FX_PATTERN_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_pattern_audio_processor_get_type ()

Types and Values

#define	AGS_FX_PATTERN_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY
#define	AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR
struct	AgsFxPatternAudioProcessor
struct	AgsFxPatternAudioProcessorClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
        &#x2570;&#x2500;&#x2500; AgsFxPatternAudioProcessor
```

Implemented Interfaces

AgsFxPatternAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_pattern_audio_processor.h>
```

Description

The AgsFxPatternAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_pattern_audio_processor_key_on ()

```
void
ags_fx_pattern_audio_processor_key_on (AgsFxPatternAudioProcessor * ↩
    fx_pattern_audio_processor,
                                       AgsNote *note,
                                       guint velocity,
                                       guint key_mode);
```

ags_fx_pattern_audio_processor_play ()

```
void
ags_fx_pattern_audio_processor_play (AgsFxPatternAudioProcessor *fx_pattern_audio_processor ↩
    );
```

ags_fx_pattern_audio_processor_counter_change ()

```
void
ags_fx_pattern_audio_processor_counter_change
                                (AgsFxPatternAudioProcessor *fx_pattern_audio_processor);
```

ags_fx_pattern_audio_processor_new ()

```
AgsFxPatternAudioProcessor~*
ags_fx_pattern_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxPatternAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxPatternAudioProcessor

Since: 3.3.0

AGS_FX_PATTERN_AUDIO_PROCESSOR()

```
#define AGS_FX_PATTERN_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR, AgsFxPatternAudioProcessor))
```

AGS_FX_PATTERN_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_PATTERN_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class) ↩
    , AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR, AgsFxPatternAudioProcessorClass))
```

AGS_FX_PATTERN_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_PATTERN_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj ↩
    ), AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR, AgsFxPatternAudioProcessorClass))
```

AGS_IS_FX_PATTERN_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_PATTERN_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR))
```

AGS_IS_FX_PATTERN_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_PATTERN_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR))
```

ags_fx_pattern_audio_processor_get_type ()

```
GType
ags_fx_pattern_audio_processor_get_type
                                ();
```

Types and Values

AGS_FX_PATTERN_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY

```
#define AGS_FX_PATTERN_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY (127)
```

AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR                ( ↩
    ags_fx_pattern_audio_processor_get_type())
```

struct AgsFxPatternAudioProcessor

```
struct AgsFxPatternAudioProcessor;
```

struct AgsFxPatternAudioProcessorClass

```
struct AgsFxPatternAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;

  void (*key_on)(AgsFxPatternAudioProcessor *fx_pattern_audio_processor,
     AgsNote *note,
     guint velocity,
     guint key_mode);

  void (*play)(AgsFxPatternAudioProcessor *fx_pattern_audio_processor);

  void (*counter_change)(AgsFxPatternAudioProcessor *fx_pattern_audio_processor);
};
```

20.3 AgsFxPatternAudioSignal

AgsFxPatternAudioSignal — fx pattern audio signal

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_fx_pattern_audio_signal_stream_feed ()
void	ags_fx_pattern_audio_signal_notify_remove ()
AgsFxPatternAudioSignal *	ags_fx_pattern_audio_signal_new ()
#define	AGS_FX_PATTERN_AUDIO_SIGNAL()
#define	AGS_FX_PATTERN_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_PATTERN_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_PATTERN_AUDIO_SIGNAL()
#define	AGS_IS_FX_PATTERN_AUDIO_SIGNAL_CLASS()
GType	ags_fx_pattern_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL
struct	AgsFxPatternAudioSignal
struct	AgsFxPatternAudioSignalClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
        &#x2570;&#x2500;&#x2500; AgsFxPatternAudioSignal
```

Implemented Interfaces

AgsFxPatternAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_pattern_audio_signal.h>
```

Description

The AgsFxPatternAudioSignal class provides ports to the effect processor.

Functions

ags_fx_pattern_audio_signal_stream_feed ()

```
void
ags_fx_pattern_audio_signal_stream_feed
                                (AgsFxPatternAudioSignal *fx_pattern_audio_signal,
                                 AgsAudioSignal *source,
                                 AgsNote *note,
                                 guint x0,
                                 guint x1,
                                 guint y,
                                 gdouble delay_counter,
                                 guint64 offset_counter,
                                 guint frame_count,
                                 gdouble delay,
                                 guint buffer_size);
```

ags_fx_pattern_audio_signal_notify_remove ()

```
void
ags_fx_pattern_audio_signal_notify_remove
                                (AgsFxPatternAudioSignal *fx_pattern_audio_signal,
                                 AgsAudioSignal *source,
                                 AgsNote *note,
                                 guint x0,
                                 guint x1,
                                 guint y);
```

ags_fx_pattern_audio_signal_new ()

```
AgsFxPatternAudioSignal~*
ags_fx_pattern_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxPatternAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxPatternAudioSignal

Since: 3.3.0

AGS_FX_PATTERN_AUDIO_SIGNAL()

```
#define AGS_FX_PATTERN_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL, AgsFxPatternAudioSignal))
```

AGS_FX_PATTERN_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_PATTERN_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL, AgsFxPatternAudioSignalClass))
```

AGS_FX_PATTERN_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_PATTERN_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL, AgsFxPatternAudioSignalClass))
```

AGS_IS_FX_PATTERN_AUDIO_SIGNAL()

```
#define AGS_IS_FX_PATTERN_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
     AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL))
```

AGS_IS_FX_PATTERN_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_PATTERN_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL))
```

ags_fx_pattern_audio_signal_get_type ()

```
GType
ags_fx_pattern_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL                ( ↩
    ags_fx_pattern_audio_signal_get_type())
```

struct AgsFxPatternAudioSignal

```
struct AgsFxPatternAudioSignal;
```

struct AgsFxPatternAudioSignalClass

```
struct AgsFxPatternAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;

  void (*stream_feed)(AgsFxPatternAudioSignal *fx_pattern_audio_signal,
          AgsAudioSignal *source,
          AgsNote *note,
          guint x0, guint x1,
          guint y,
          gdouble delay_counter, guint64 offset_counter,
          guint frame_count,
```

```
          gdouble delay, guint buffer_size);

  void (*notify_remove)(AgsFxPatternAudioSignal *fx_pattern_audio_signal,
      AgsAudioSignal *source,
      AgsNote *note,
      guint x0, guint x1,
      guint y);
};
```

20.4 AgsFxPatternChannel

AgsFxPatternChannel — fx pattern channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPatternChannel *	ags_fx_pattern_channel_new ()
#define	AGS_FX_PATTERN_CHANNEL()
#define	AGS_FX_PATTERN_CHANNEL_CLASS()
#define	AGS_FX_PATTERN_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_PATTERN_CHANNEL()
#define	AGS_IS_FX_PATTERN_CHANNEL_CLASS()
GType	ags_fx_pattern_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_PATTERN_CHANNEL
struct	AgsFxPatternChannel
struct	AgsFxPatternChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxPatternChannel
```

Implemented Interfaces

AgsFxPatternChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_pattern_channel.h>
```

Description

The AgsFxPatternChannel class provides ports to the effect processor.

Functions

ags_fx_pattern_channel_new ()

```
AgsFxPatternChannel~*
ags_fx_pattern_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxPatternChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxPatternChannel

Since: 3.3.0

AGS_FX_PATTERN_CHANNEL()

```
#define AGS_FX_PATTERN_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PATTERN_CHANNEL, AgsFxPatternChannel))
```

AGS_FX_PATTERN_CHANNEL_CLASS()

```
#define AGS_FX_PATTERN_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PATTERN_CHANNEL, AgsFxPatternChannelClass))
```

AGS_FX_PATTERN_CHANNEL_GET_CLASS()

```
#define AGS_FX_PATTERN_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PATTERN_CHANNEL, AgsFxPatternChannelClass))
```

AGS_IS_FX_PATTERN_CHANNEL()

```
#define AGS_IS_FX_PATTERN_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PATTERN_CHANNEL))
```

AGS_IS_FX_PATTERN_CHANNEL_CLASS()

```
#define AGS_IS_FX_PATTERN_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PATTERN_CHANNEL))
```

ags_fx_pattern_channel_get_type ()

```
GType
ags_fx_pattern_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_PATTERN_CHANNEL

```
#define AGS_TYPE_FX_PATTERN_CHANNEL                (ags_fx_pattern_channel_get_type())
```

struct AgsFxPatternChannel

```
struct AgsFxPatternChannel;
```

struct AgsFxPatternChannelClass

```
struct AgsFxPatternChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

20.5 AgsFxPatternChannelProcessor

AgsFxPatternChannelProcessor — fx pattern channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPatternChannelProcessor *	ags_fx_pattern_channel_processor_new ()
#define	AGS_FX_PATTERN_CHANNEL_PROCESSOR()
#define	AGS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_PATTERN_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_pattern_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR
struct	AgsFxPatternChannelProcessor
struct	AgsFxPatternChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxPatternChannelProcessor
```

Implemented Interfaces

AgsFxPatternChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_pattern_channel_processor.h>
```

Description

The AgsFxPatternChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_pattern_channel_processor_new ()

```
AgsFxPatternChannelProcessor~*
ags_fx_pattern_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxPatternChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxPatternChannelProcessor

Since: 3.3.0

AGS_FX_PATTERN_CHANNEL_PROCESSOR()

```
#define AGS_FX_PATTERN_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR, AgsFxPatternChannelProcessor))
```

AGS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR, AgsFxPatternChannelProcessorClass))
```

AGS_FX_PATTERN_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_PATTERN_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR, AgsFxPatternChannelProcessorClass))
```

AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR))
```

AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR))
```

ags_fx_pattern_channel_processor_get_type ()

```
GType
ags_fx_pattern_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR                ( ↩
    ags_fx_pattern_channel_processor_get_type())
```

struct AgsFxPatternChannelProcessor

```
struct AgsFxPatternChannelProcessor;
```

struct AgsFxPatternChannelProcessorClass

```
struct AgsFxPatternChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

20.6 AgsFxPatternRecycling

AgsFxPatternRecycling — fx pattern recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPatternRecycling *	ags_fx_pattern_recycling_new ()
#define	AGS_FX_PATTERN_RECYCLING()
#define	AGS_FX_PATTERN_RECYCLING_CLASS()
#define	AGS_FX_PATTERN_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_PATTERN_RECYCLING()
#define	AGS_IS_FX_PATTERN_RECYCLING_CLASS()
GType	ags_fx_pattern_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_PATTERN_RECYCLING
struct	AgsFxPatternRecycling
struct	AgsFxPatternRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxPatternRecycling
```

Implemented Interfaces

AgsFxPatternRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_pattern_recycling.h>
```

Description

The AgsFxPatternRecycling class provides ports to the effect processor.

Functions

ags_fx_pattern_recycling_new ()

```
AgsFxPatternRecycling~*
ags_fx_pattern_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxPatternRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxPatternRecycling

Since: 3.3.0

AGS_FX_PATTERN_RECYCLING()

```
#define AGS_FX_PATTERN_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PATTERN_RECYCLING, AgsFxPatternRecycling))
```

AGS_FX_PATTERN_RECYCLING_CLASS()

```
#define AGS_FX_PATTERN_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PATTERN_RECYCLING, AgsFxPatternRecyclingClass))
```

AGS_FX_PATTERN_RECYCLING_GET_CLASS()

```
#define AGS_FX_PATTERN_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PATTERN_RECYCLING, AgsFxPatternRecyclingClass))
```

AGS_IS_FX_PATTERN_RECYCLING()

```
#define AGS_IS_FX_PATTERN_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PATTERN_RECYCLING))
```

AGS_IS_FX_PATTERN_RECYCLING_CLASS()

```
#define AGS_IS_FX_PATTERN_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PATTERN_RECYCLING))
```

ags_fx_pattern_recycling_get_type ()

```
GType
ags_fx_pattern_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_PATTERN_RECYCLING

```
#define AGS_TYPE_FX_PATTERN_RECYCLING                (ags_fx_pattern_recycling_get_type())
```

struct AgsFxPatternRecycling

```
struct AgsFxPatternRecycling;
```

struct AgsFxPatternRecyclingClass

```
struct AgsFxPatternRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

20.7 AgsFxNotationAudio

AgsFxNotationAudio — fx notation audio

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_fx_notation_audio_test_flags ()
void	ags_fx_notation_audio_set_flags ()
void	ags_fx_notation_audio_unset_flags ()
gboolean	ags_fx_notation_audio_get_pattern_mode ()
void	ags_fx_notation_audio_set_pattern_mode ()
GList *	ags_fx_notation_audio_get_feed_note ()
void	ags_fx_notation_audio_add_feed_note ()
void	ags_fx_notation_audio_remove_feed_note ()
AgsFxNotationAudio *	ags_fx_notation_audio_new ()
#define	AGS_FX_NOTATION_AUDIO()
#define	AGS_FX_NOTATION_AUDIO_CLASS()
#define	AGS_FX_NOTATION_AUDIO_GET_CLASS()
#define	AGS_IS_FX_NOTATION_AUDIO()
#define	AGS_IS_FX_NOTATION_AUDIO_CLASS()
GType	ags_fx_notation_audio_get_type ()

Properties

AgsPort *	bpm	Read / Write
AgsPort *	delay	Read / Write
AgsPort *	duration	Read / Write
AgsPort *	loop	Read / Write
AgsPort *	loop-end	Read / Write
AgsPort *	loop-start	Read / Write
AgsPort *	tact	Read / Write

Types and Values

#define	AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_START
#define	AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_END
enum	AgsFxNotationAudioFlags
#define	AGS_TYPE_FX_NOTATION_AUDIO
struct	AgsFxNotationAudio

struct	AgsFxNotationAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudio
                &#x251c;&#x2500;&#x2500; AgsFxDssiAudio
                &#x251c;&#x2500;&#x2500; AgsFxFMSynthAudio
                &#x251c;&#x2500;&#x2500; AgsFxLv2Audio
                &#x251c;&#x2500;&#x2500; AgsFxSF2SynthAudio
                &#x251c;&#x2500;&#x2500; AgsFxSFZSynthAudio
                &#x2570;&#x2500;&#x2500; AgsFxSynthAudio
```

Implemented Interfaces

AgsFxNotationAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_notation_audio.h>
```

Description

The AgsFxNotationAudio class provides ports to the effect processor.

Functions

ags_fx_notation_audio_test_flags ()

```
gboolean
ags_fx_notation_audio_test_flags (AgsFxNotationAudio *fx_notation_audio,
                                  guint flags);
```

Test *flags* of *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	
flags	the flags	

Returns

TRUE on success, otherwise FALSE

Since: 3.3.0

ags_fx_notation_audio_set_flags ()

```
void
ags_fx_notation_audio_set_flags (AgsFxNotationAudio *fx_notation_audio,
                                 guint flags);
```

Set *flags* of *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	
flags	the flags	

Since: 3.3.0

ags_fx_notation_audio_unset_flags ()

```
void
ags_fx_notation_audio_unset_flags (AgsFxNotationAudio *fx_notation_audio,
                                   guint flags);
```

Unset *flags* of *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	
flags	the flags	

Since: 3.3.0

ags_fx_notation_audio_get_pattern_mode ()

```
gboolean
ags_fx_notation_audio_get_pattern_mode
                                (AgsFxNotationAudio *fx_notation_audio);
```

Get pattern mode of *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	

Returns

TRUE if pattern mode, otherwise FALSE

Since: 3.3.0

ags_fx_notation_audio_set_pattern_mode ()

```
void
ags_fx_notation_audio_set_pattern_mode
                                (AgsFxNotationAudio *fx_notation_audio,
                                 gboolean pattern_mode);
```

Set *pattern_mode* of *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	
pattern_mode	TRUE if pattern mode, otherwise FALSE	

Since: 3.3.0

ags_fx_notation_audio_get_feed_note ()

```
GList~*
ags_fx_notation_audio_get_feed_note (AgsFxNotationAudio *fx_notation_audio);
```

Get feed note of *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	

Returns

the GList containing feed note.

[element-type AgsAudio.Note][transfer full]

Since: 3.3.0

ags_fx_notation_audio_add_feed_note ()

```
void
ags_fx_notation_audio_add_feed_note (AgsFxNotationAudio *fx_notation_audio,
                                     AgsNote *note);
```

Add *note* to *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	
note	the AgsNote	

Since: 3.3.0

ags_fx_notation_audio_remove_feed_note ()

```
void
ags_fx_notation_audio_remove_feed_note
                                (AgsFxNotationAudio *fx_notation_audio,
                                 AgsNote *note);
```

Remove *note* from *fx_notation_audio*.

Parameters

fx_notation_audio	the AgsFxNotationAudio	
note	the AgsNote	

Since: 3.3.0

ags_fx_notation_audio_new ()

```
AgsFxNotationAudio~*
ags_fx_notation_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxNotationAudio

Parameters

audio	the AgsAudio	

Returns

the new AgsFxNotationAudio

Since: 3.3.0

AGS_FX_NOTATION_AUDIO()

```
#define AGS_FX_NOTATION_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_NOTATION_AUDIO, AgsFxNotationAudio))
```

AGS_FX_NOTATION_AUDIO_CLASS()

```
#define AGS_FX_NOTATION_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_NOTATION_AUDIO, AgsFxNotationAudioClass))
```

AGS_FX_NOTATION_AUDIO_GET_CLASS()

```
#define AGS_FX_NOTATION_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_NOTATION_AUDIO, AgsFxNotationAudioClass))
```

AGS_IS_FX_NOTATION_AUDIO()

```
#define AGS_IS_FX_NOTATION_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_NOTATION_AUDIO))
```

AGS_IS_FX_NOTATION_AUDIO_CLASS()

```
#define AGS_IS_FX_NOTATION_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_NOTATION_AUDIO))
```

ags_fx_notation_audio_get_type ()

```
GType
ags_fx_notation_audio_get_type ();
```

Types and Values

AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_START

```
#define AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_START (0)
```

AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_END

```
#define AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_END (64)
```

enum AgsFxNotationAudioFlags

Enum values to enable specific feature of AgsFxNotationAudio.

Members

AGS_FX_NOTATION_AUDIO_PLAY	do play notation
AGS_FX_NOTATION_AUDIO_RECORD	do record notation
AGS_FX_NOTATION_AUDIO_FEED	do feed note

AGS_TYPE_FX_NOTATION_AUDIO

```
#define AGS_TYPE_FX_NOTATION_AUDIO                 (ags_fx_notation_audio_get_type())
```

struct AgsFxNotationAudio

```
struct AgsFxNotationAudio;
```

struct AgsFxNotationAudioClass

```
struct AgsFxNotationAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

Property Details

The "bpm" property

```
  "bpm"                       AgsPort~*
```

The beats per minute.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

The "delay" property

```
  "delay"                     AgsPort~*
```

The notation's delay.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

The "duration" property

```
  "duration"                  AgsPort~*
```

The notation's duration.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

The "loop" property

```
  "loop"                      AgsPort~*
```

Do loop.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

The "loop-end" property

```
"loop-end"                  AgsPort~*
```

The notation's loop-end.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

The "loop-start" property

```
"loop-start"                AgsPort~*
```

The notation's loop-start.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

The "tact" property

```
"tact"                      AgsPort~*
```

The tact segmentation.

Owner: AgsFxNotationAudio

Flags: Read / Write

Since: 3.3.0

20.8 AgsFxNotationAudioProcessor

AgsFxNotationAudioProcessor — fx notation audio processor

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_fx_notation_audio_processor_key_on ()
void	ags_fx_notation_audio_processor_key_off ()
void	ags_fx_notation_audio_processor_key_pressure ()
void	ags_fx_notation_audio_processor_play ()
void	ags_fx_notation_audio_processor_record ()
void	ags_fx_notation_audio_processor_feed ()
void	ags_fx_notation_audio_processor_counter_change ()
AgsFxNotationAudioProcessor *	ags_fx_notation_audio_processor_new ()
#define	AGS_FX_NOTATION_AUDIO_PROCESSOR()

#define	AGS_FX_NOTATION_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_NOTATION_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_NOTATION_AUDIO_PROCESSOR()
#define	AGS_IS_FX_NOTATION_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_notation_audio_processor_get_type ()

Types and Values

#define	AGS_FX_NOTATION_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY
enum	AgsFxNotationAudioProcessorKeyMode
#define	AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR
struct	AgsFxNotationAudioProcessor
struct	AgsFxNotationAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioProcessor
                &#x251c;&#x2500;&#x2500; AgsFxDssiAudioProcessor
                &#x251c;&#x2500;&#x2500; AgsFxFMSynthAudioProcessor
                &#x251c;&#x2500;&#x2500; AgsFxLv2AudioProcessor
                &#x251c;&#x2500;&#x2500; AgsFxSF2SynthAudioProcessor
                &#x251c;&#x2500;&#x2500; AgsFxSFZSynthAudioProcessor
                &#x2570;&#x2500;&#x2500; AgsFxSynthAudioProcessor
```

Implemented Interfaces

AgsFxNotationAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_notation_audio_processor.h>
```

Description

The AgsFxNotationAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_notation_audio_processor_key_on ()

```
void
ags_fx_notation_audio_processor_key_on
                                (AgsFxNotationAudioProcessor *fx_notation_audio_processor,
                                 AgsNote *note,
                                 guint velocity,
                                 guint key_mode);
```

ags_fx_notation_audio_processor_key_off ()

```
void
ags_fx_notation_audio_processor_key_off
                                (AgsFxNotationAudioProcessor *fx_notation_audio_processor,
                                 AgsNote *note,
                                 guint velocity,
                                 guint key_mode);
```

ags_fx_notation_audio_processor_key_pressure ()

```
void
ags_fx_notation_audio_processor_key_pressure
                                (AgsFxNotationAudioProcessor *fx_notation_audio_processor,
                                 AgsNote *note,
                                 guint velocity,
                                 guint key_mode);
```

ags_fx_notation_audio_processor_play ()

```
void
ags_fx_notation_audio_processor_play (AgsFxNotationAudioProcessor * ↩
    fx_notation_audio_processor);
```

ags_fx_notation_audio_processor_record ()

```
void
ags_fx_notation_audio_processor_record
                                (AgsFxNotationAudioProcessor *fx_notation_audio_processor);
```

ags_fx_notation_audio_processor_feed ()

```
void
ags_fx_notation_audio_processor_feed (AgsFxNotationAudioProcessor * ↩
    fx_notation_audio_processor);
```

ags_fx_notation_audio_processor_counter_change ()

```
void
ags_fx_notation_audio_processor_counter_change
                                (AgsFxNotationAudioProcessor *fx_notation_audio_processor);
```

ags_fx_notation_audio_processor_new ()

```
AgsFxNotationAudioProcessor~*
ags_fx_notation_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxNotationAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxNotationAudioProcessor

Since: 3.3.0

AGS_FX_NOTATION_AUDIO_PROCESSOR()

```
#define AGS_FX_NOTATION_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR, AgsFxNotationAudioProcessor))
```

AGS_FX_NOTATION_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_NOTATION_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR, AgsFxNotationAudioProcessorClass))
```

AGS_FX_NOTATION_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_NOTATION_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR, AgsFxNotationAudioProcessorClass))
```

AGS_IS_FX_NOTATION_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_NOTATION_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR))
```

AGS_IS_FX_NOTATION_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_NOTATION_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR))
```

ags_fx_notation_audio_processor_get_type ()

```
GType
ags_fx_notation_audio_processor_get_type
                                ();
```

Types and Values

AGS_FX_NOTATION_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY

```
#define AGS_FX_NOTATION_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY (127)
```

enum AgsFxNotationAudioProcessorKeyMode

Members

AGS_FX_NOTATION_AUDIO_PROCESSOR_KEY_MODE_PLAY	
AGS_FX_NOTATION_AUDIO_PROCESSOR_KEY_MODE_RECORD	
AGS_FX_NOTATION_AUDIO_PROCESSOR_KEY_MODE_FEED	

AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR                ( ↩
    ags_fx_notation_audio_processor_get_type())
```

struct AgsFxNotationAudioProcessor

```
struct AgsFxNotationAudioProcessor;
```

struct AgsFxNotationAudioProcessorClass

```
struct AgsFxNotationAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;

  void (*key_on)(AgsFxNotationAudioProcessor *fx_notation_audio_processor,
     AgsNote *note,
     guint velocity,
     guint key_mode);
  void (*key_off)(AgsFxNotationAudioProcessor *fx_notation_audio_processor,
      AgsNote *note,
      guint velocity,
      guint key_mode);
  void (*key_pressure)(AgsFxNotationAudioProcessor *fx_notation_audio_processor,
           AgsNote *note,
           guint velocity,
           guint key_mode);

  void (*play)(AgsFxNotationAudioProcessor *fx_notation_audio_processor);
  void (*record)(AgsFxNotationAudioProcessor *fx_notation_audio_processor);
  void (*feed)(AgsFxNotationAudioProcessor *fx_notation_audio_processor);

  void (*counter_change)(AgsFxNotationAudioProcessor *fx_notation_audio_processor);
};
```

20.9 AgsFxNotationAudioSignal

AgsFxNotationAudioSignal — fx notation audio signal

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_fx_notation_audio_signal_stream_feed ()
void	ags_fx_notation_audio_signal_notify_remove ()
AgsFxNotationAudioSignal *	ags_fx_notation_audio_signal_new ()
#define	AGS_FX_NOTATION_AUDIO_SIGNAL()
#define	AGS_FX_NOTATION_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_NOTATION_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_NOTATION_AUDIO_SIGNAL()
#define	AGS_IS_FX_NOTATION_AUDIO_SIGNAL_CLASS()
GType	ags_fx_notation_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL
struct	AgsFxNotationAudioSignal
struct	AgsFxNotationAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioSignal
                &#x251c;&#x2500;&#x2500; AgsFxDssiAudioSignal
                &#x251c;&#x2500;&#x2500; AgsFxFMSynthAudioSignal
                &#x251c;&#x2500;&#x2500; AgsFxLv2AudioSignal
                &#x251c;&#x2500;&#x2500; AgsFxSF2SynthAudioSignal
                &#x251c;&#x2500;&#x2500; AgsFxSFZSynthAudioSignal
                &#x2570;&#x2500;&#x2500; AgsFxSynthAudioSignal
```

Implemented Interfaces

AgsFxNotationAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_notation_audio_signal.h>
```

Description

The AgsFxNotationAudioSignal class provides ports to the effect processor.

Functions

ags_fx_notation_audio_signal_stream_feed ()

```
void
ags_fx_notation_audio_signal_stream_feed
                                (AgsFxNotationAudioSignal *fx_notation_audio_signal,
                                 AgsAudioSignal *source,
                                 AgsNote *note,
                                 gboolean pattern_mode,
                                 guint x0,
                                 guint x1,
                                 guint y,
                                 gdouble delay_counter,
                                 guint64 offset_counter,
                                 guint frame_count,
                                 gdouble delay,
                                 guint buffer_size);
```

ags_fx_notation_audio_signal_notify_remove ()

```
void
ags_fx_notation_audio_signal_notify_remove
                                (AgsFxNotationAudioSignal *fx_notation_audio_signal,
                                 AgsAudioSignal *source,
                                 AgsNote *note,
                                 guint x0,
                                 guint x1,
                                 guint y);
```

ags_fx_notation_audio_signal_new ()

```
AgsFxNotationAudioSignal~*
ags_fx_notation_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxNotationAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxNotationAudioSignal

Since: 3.3.0

AGS_FX_NOTATION_AUDIO_SIGNAL()

```
#define AGS_FX_NOTATION_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL, AgsFxNotationAudioSignal))
```

AGS_FX_NOTATION_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_NOTATION_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL, AgsFxNotationAudioSignalClass))
```

AGS_FX_NOTATION_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_NOTATION_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
    AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL, AgsFxNotationAudioSignalClass))
```

AGS_IS_FX_NOTATION_AUDIO_SIGNAL()

```
#define AGS_IS_FX_NOTATION_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
   , AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL))
```

AGS_IS_FX_NOTATION_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_NOTATION_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
    AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL))
```

ags_fx_notation_audio_signal_get_type ()

```
GType
ags_fx_notation_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL                ( ↩
   ags_fx_notation_audio_signal_get_type())
```

struct AgsFxNotationAudioSignal

```
struct AgsFxNotationAudioSignal;
```

struct AgsFxNotationAudioSignalClass

```
struct AgsFxNotationAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;

  void (*stream_feed)(AgsFxNotationAudioSignal *fx_notation_audio_signal,
          AgsAudioSignal *source,
          AgsNote *note,
          gboolean pattern_mode,
          guint x0, guint x1,
          guint y,
          gdouble delay_counter, guint64 offset_counter,
          guint frame_count,
          gdouble delay, guint buffer_size);

  void (*notify_remove)(AgsFxNotationAudioSignal *fx_notation_audio_signal,
      AgsAudioSignal *source,
      AgsNote *note,
      guint x0, guint x1,
      guint y);
};
```

20.10 AgsFxNotationChannel

AgsFxNotationChannel — fx notation channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxNotationChannel *	ags_fx_notation_channel_new ()
#define	AGS_FX_NOTATION_CHANNEL()
#define	AGS_FX_NOTATION_CHANNEL_CLASS()
#define	AGS_FX_NOTATION_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_NOTATION_CHANNEL()
#define	AGS_IS_FX_NOTATION_CHANNEL_CLASS()
GType	ags_fx_notation_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_NOTATION_CHANNEL
struct	AgsFxNotationChannel
struct	AgsFxNotationChannelClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannel
        &#x2570;&#x2500;&#x2500; AgsFxNotationChannel
            &#x251c;&#x2500;&#x2500; AgsFxDssiChannel
            &#x251c;&#x2500;&#x2500; AgsFxFMSynthChannel
            &#x251c;&#x2500;&#x2500; AgsFxLv2Channel
            &#x251c;&#x2500;&#x2500; AgsFxSF2SynthChannel
            &#x251c;&#x2500;&#x2500; AgsFxSFZSynthChannel
            &#x2570;&#x2500;&#x2500; AgsFxSynthChannel
```

Implemented Interfaces

AgsFxNotationChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_notation_channel.h>
```

Description

The AgsFxNotationChannel class provides ports to the effect processor.

Functions

ags_fx_notation_channel_new ()

```
AgsFxNotationChannel~*
ags_fx_notation_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxNotationChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxNotationChannel

Since: 3.3.0

AGS_FX_NOTATION_CHANNEL()

```
#define AGS_FX_NOTATION_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_NOTATION_CHANNEL, AgsFxNotationChannel))
```

AGS_FX_NOTATION_CHANNEL_CLASS()

```
#define AGS_FX_NOTATION_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_NOTATION_CHANNEL, AgsFxNotationChannelClass))
```

AGS_FX_NOTATION_CHANNEL_GET_CLASS()

```
#define AGS_FX_NOTATION_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_NOTATION_CHANNEL, AgsFxNotationChannelClass))
```

AGS_IS_FX_NOTATION_CHANNEL()

```
#define AGS_IS_FX_NOTATION_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_NOTATION_CHANNEL))
```

AGS_IS_FX_NOTATION_CHANNEL_CLASS()

```
#define AGS_IS_FX_NOTATION_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_NOTATION_CHANNEL))
```

ags_fx_notation_channel_get_type ()

```
GType
ags_fx_notation_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_NOTATION_CHANNEL

```
#define AGS_TYPE_FX_NOTATION_CHANNEL                (ags_fx_notation_channel_get_type())
```

struct AgsFxNotationChannel

```
struct AgsFxNotationChannel;
```

struct AgsFxNotationChannelClass

```
struct AgsFxNotationChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

20.11 AgsFxNotationChannelProcessor

AgsFxNotationChannelProcessor — fx notation channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxNotationChannelProcessor *	ags_fx_notation_channel_processor_new ()
#define	AGS_FX_NOTATION_CHANNEL_PROCESSOR()
#define	AGS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_NOTATION_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_notation_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR
struct	AgsFxNotationChannelProcessor
struct	AgsFxNotationChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannelProcessor
                &#x251c;&#x2500;&#x2500; AgsFxDssiChannelProcessor
                &#x251c;&#x2500;&#x2500; AgsFxFMSynthChannelProcessor
```

```
                &#x251c;&#x2500;&#x2500; AgsFxLv2ChannelProcessor
                &#x251c;&#x2500;&#x2500; AgsFxSF2SynthChannelProcessor
                &#x251c;&#x2500;&#x2500; AgsFxSFZSynthChannelProcessor
                &#x2570;&#x2500;&#x2500; AgsFxSynthChannelProcessor
```

Implemented Interfaces

AgsFxNotationChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_notation_channel_processor.h>
```

Description

The AgsFxNotationChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_notation_channel_processor_new ()

```
AgsFxNotationChannelProcessor~*
ags_fx_notation_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxNotationChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxNotationChannelProcessor

Since: 3.3.0

AGS_FX_NOTATION_CHANNEL_PROCESSOR()

```
#define AGS_FX_NOTATION_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR, AgsFxNotationChannelProcessor))
```

AGS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR, AgsFxNotationChannelProcessorClass))
```

AGS_FX_NOTATION_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_NOTATION_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR, AgsFxNotationChannelProcessorClass))
```

AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR))
```

AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR))
```

ags_fx_notation_channel_processor_get_type ()

```
GType
ags_fx_notation_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR                ( ↩
    ags_fx_notation_channel_processor_get_type())
```

struct AgsFxNotationChannelProcessor

```
struct AgsFxNotationChannelProcessor;
```

struct AgsFxNotationChannelProcessorClass

```
struct AgsFxNotationChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

20.12 AgsFxNotationRecycling

AgsFxNotationRecycling — fx notation recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxNotationRecycling *	ags_fx_notation_recycling_new ()
#define	AGS_FX_NOTATION_RECYCLING()
#define	AGS_FX_NOTATION_RECYCLING_CLASS()
#define	AGS_FX_NOTATION_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_NOTATION_RECYCLING()
#define	AGS_IS_FX_NOTATION_RECYCLING_CLASS()
GType	ags_fx_notation_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_NOTATION_RECYCLING
struct	AgsFxNotationRecycling
struct	AgsFxNotationRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxNotationRecycling
                &#x251c;&#x2500;&#x2500; AgsFxDssiRecycling
                &#x251c;&#x2500;&#x2500; AgsFxFMSynthRecycling
                &#x251c;&#x2500;&#x2500; AgsFxLv2Recycling
                &#x251c;&#x2500;&#x2500; AgsFxSF2SynthRecycling
                &#x251c;&#x2500;&#x2500; AgsFxSFZSynthRecycling
                &#x2570;&#x2500;&#x2500; AgsFxSynthRecycling
```

Implemented Interfaces

AgsFxNotationRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_notation_recycling.h>
```

Description

The AgsFxNotationRecycling class provides ports to the effect processor.

Functions

ags_fx_notation_recycling_new ()

```
AgsFxNotationRecycling~*
ags_fx_notation_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxNotationRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxNotationRecycling

Since: 3.3.0

AGS_FX_NOTATION_RECYCLING()

```
#define AGS_FX_NOTATION_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_NOTATION_RECYCLING, AgsFxNotationRecycling))
```

AGS_FX_NOTATION_RECYCLING_CLASS()

```
#define AGS_FX_NOTATION_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_NOTATION_RECYCLING, AgsFxNotationRecyclingClass))
```

AGS_FX_NOTATION_RECYCLING_GET_CLASS()

```
#define AGS_FX_NOTATION_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_NOTATION_RECYCLING, AgsFxNotationRecyclingClass))
```

AGS_IS_FX_NOTATION_RECYCLING()

```
#define AGS_IS_FX_NOTATION_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_NOTATION_RECYCLING))
```

AGS_IS_FX_NOTATION_RECYCLING_CLASS()

```
#define AGS_IS_FX_NOTATION_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_NOTATION_RECYCLING))
```

ags_fx_notation_recycling_get_type ()

```
GType
ags_fx_notation_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_NOTATION_RECYCLING

```
#define AGS_TYPE_FX_NOTATION_RECYCLING                (ags_fx_notation_recycling_get_type() ↩
    )
```

struct AgsFxNotationRecycling

```
struct AgsFxNotationRecycling;
```

struct AgsFxNotationRecyclingClass

```
struct AgsFxNotationRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

Plugins extending ags-fx-notation.

20.13 AgsFxDssiAudio

AgsFxDssiAudio — fx dssi audio

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_DSSI_AUDIO_SCOPE_DATA()
#define	AGS_FX_DSSI_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX()
#define	AGS_FX_DSSI_AUDIO_CHANNEL_DATA()
#define	AGS_FX_DSSI_AUDIO_CHANNEL_DATA_GET_STRCT_MUTEX()
#define	AGS_FX_DSSI_AUDIO_INPUT_DATA()
#define	AGS_FX_DSSI_AUDIO_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxDssiAudioScopeData *	ags_fx_dssi_audio_scope_data_alloc ()
void	ags_fx_dssi_audio_scope_data_free ()
AgsFxDssiAudioChannelData *	ags_fx_dssi_audio_channel_data_alloc ()
void	ags_fx_dssi_audio_channel_data_free ()
AgsFxDssiAudioInputData *	ags_fx_dssi_audio_input_data_alloc ()
void	ags_fx_dssi_audio_input_data_free ()
gboolean	ags_fx_dssi_audio_test_flags ()
void	ags_fx_dssi_audio_set_flags ()
void	ags_fx_dssi_audio_unset_flags ()
void	ags_fx_dssi_audio_load_plugin ()
void	ags_fx_dssi_audio_load_port ()
void	ags_fx_dssi_audio_change_program ()
AgsFxDssiAudio *	ags_fx_dssi_audio_new ()
#define	AGS_FX_DSSI_AUDIO()
#define	AGS_FX_DSSI_AUDIO_CLASS()
#define	AGS_FX_DSSI_AUDIO_GET_CLASS()
#define	AGS_IS_FX_DSSI_AUDIO()
#define	AGS_IS_FX_DSSI_AUDIO_CLASS()
GType	ags_fx_dssi_audio_get_type ()

Types and Values

enum	AgsFxDssiAudioFlags
struct	AgsFxDssiAudioScopeData
struct	AgsFxDssiAudioChannelData
struct	AgsFxDssiAudioInputData
#define	AGS_TYPE_FX_DSSI_AUDIO
struct	AgsFxDssiAudio
struct	AgsFxDssiAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudio
                &#x2570;&#x2500;&#x2500; AgsFxDssiAudio
```

Implemented Interfaces

AgsFxDssiAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_dssi_audio.h>
```

Description

The AgsFxDssiAudio class provides ports to the effect processor.

Functions

AGS_FX_DSSI_AUDIO_SCOPE_DATA()

```
#define AGS_FX_DSSI_AUDIO_SCOPE_DATA(ptr) ((AgsFxDssiAudioScopeData *) (ptr))
```

AGS_FX_DSSI_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_DSSI_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxDssiAudioScopeData *)( ↩
    ptr))->strct_mutex))
```

AGS_FX_DSSI_AUDIO_CHANNEL_DATA()

```
#define AGS_FX_DSSI_AUDIO_CHANNEL_DATA(ptr) ((AgsFxDssiAudioChannelData *) (ptr))
```

AGS_FX_DSSI_AUDIO_CHANNEL_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_DSSI_AUDIO_CHANNEL_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxDssiAudioChannelData ↩
    *)(ptr))->strct_mutex))
```

AGS_FX_DSSI_AUDIO_INPUT_DATA()

```
#define AGS_FX_DSSI_AUDIO_INPUT_DATA(ptr) ((AgsFxDssiAudioInputData *) (ptr))
```

AGS_FX_DSSI_AUDIO_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_DSSI_AUDIO_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxDssiAudioInputData *)( ↩
    ptr))->strct_mutex))
```

ags_fx_dssi_audio_scope_data_alloc ()

```
AgsFxDssiAudioScopeData~*
ags_fx_dssi_audio_scope_data_alloc ();
```

Allocate AgsFxDssiAudioScopeData

Returns

the new AgsFxDssiAudioScopeData

Since: 3.3.0

ags_fx_dssi_audio_scope_data_free ()

```
void
ags_fx_dssi_audio_scope_data_free (AgsFxDssiAudioScopeData *scope_data);
```

Free *scope_data* .

Parameters

scope_data	the AgsFxDssiAudioScopeData

Since: 3.3.0

ags_fx_dssi_audio_channel_data_alloc ()

```
AgsFxDssiAudioChannelData~*
ags_fx_dssi_audio_channel_data_alloc ();
```

Allocate AgsFxDssiAudioChannelData

Returns

the new AgsFxDssiAudioChannelData

Since: 3.3.0

ags_fx_dssi_audio_channel_data_free ()

```
void
ags_fx_dssi_audio_channel_data_free (AgsFxDssiAudioChannelData *channel_data);
```

Free *channel_data*.

Parameters

channel_data	the AgsFxDssiAudioChannelData

Since: 3.3.0

ags_fx_dssi_audio_input_data_alloc ()

```
AgsFxDssiAudioInputData~*
ags_fx_dssi_audio_input_data_alloc ();
```

Allocate AgsFxDssiAudioInputData

Returns

the new AgsFxDssiAudioInputData

Since: 3.3.0

ags_fx_dssi_audio_input_data_free ()

```
void
ags_fx_dssi_audio_input_data_free (AgsFxDssiAudioInputData *input_data);
```

Free *input_data*.

Parameters

input_data	the AgsFxDssiAudioInputData

Since: 3.3.0

ags_fx_dssi_audio_test_flags ()

```
gboolean
ags_fx_dssi_audio_test_flags (AgsFxDssiAudio *fx_dssi_audio,
                              guint flags);
```

Test *flags* of *fx_dssi_audio*.

Parameters

fx_dssi_audio	the AgsFxDssiAudio	
flags	the flags	

Returns

TRUE on success, otherwise FALSE

Since: 3.3.0

ags_fx_dssi_audio_set_flags ()

```
void
ags_fx_dssi_audio_set_flags (AgsFxDssiAudio *fx_dssi_audio,
                             guint flags);
```

Set *flags* of *fx_dssi_audio* .

Parameters

fx_dssi_audio	the AgsFxDssiAudio	
flags	the flags	

Since: 3.3.0

ags_fx_dssi_audio_unset_flags ()

```
void
ags_fx_dssi_audio_unset_flags (AgsFxDssiAudio *fx_dssi_audio,
                               guint flags);
```

Unset *flags* of *fx_dssi_audio* .

Parameters

fx_dssi_audio	the AgsFxDssiAudio	
flags	the flags	

Since: 3.3.0

ags_fx_dssi_audio_load_plugin ()

```
void
ags_fx_dssi_audio_load_plugin (AgsFxDssiAudio *fx_dssi_audio);
```

Load plugin of *fx_dssi_audio* .

Parameters

fx_dssi_audio	the AgsFxDssiAudio

Since: 3.3.0

ags_fx_dssi_audio_load_port ()

```
void
ags_fx_dssi_audio_load_port (AgsFxDssiAudio *fx_dssi_audio);
```

Load port of *fx_dssi_audio* .

Parameters

fx_dssi_audio	the AgsFxDssiAudio

Since: 3.3.0

ags_fx_dssi_audio_change_program ()

```
void
ags_fx_dssi_audio_change_program (AgsFxDssiAudio *fx_dssi_audio,
                                  guint bank_index,
                                  guint program_index);
```

Change program of *fx_dssi_audio* .

Parameters

fx_dssi_audio	the AgsFxDssiAudio
bank_index	the bank index
program_index	the program index

Since: 3.3.0

ags_fx_dssi_audio_new ()

```
AgsFxDssiAudio~*
ags_fx_dssi_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxDssiAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxDssiAudio

Since: 3.3.0

AGS_FX_DSSI_AUDIO()

```
#define AGS_FX_DSSI_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO, AgsFxDssiAudio))
```

AGS_FX_DSSI_AUDIO_CLASS()

```
#define AGS_FX_DSSI_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_DSSI_AUDIO, AgsFxDssiAudioClass))
```

AGS_FX_DSSI_AUDIO_GET_CLASS()

```
#define AGS_FX_DSSI_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO, AgsFxDssiAudioClass))
```

AGS_IS_FX_DSSI_AUDIO()

```
#define AGS_IS_FX_DSSI_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO))
```

AGS_IS_FX_DSSI_AUDIO_CLASS()

```
#define AGS_IS_FX_DSSI_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_DSSI_AUDIO))
```

ags_fx_dssi_audio_get_type ()

```
GType
ags_fx_dssi_audio_get_type ();
```

Types and Values

enum AgsFxDssiAudioFlags

Members

AGS_FX_DSSI_AUDIO_LIVE_INSTRUMENT	

struct AgsFxDssiAudioScopeData

```
struct AgsFxDssiAudioScopeData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint audio_channels;

  AgsFxDssiAudioChannelData **channel_data;
};
```

struct AgsFxDssiAudioChannelData

```
struct AgsFxDssiAudioChannelData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint event_count;

  LADSPA_Data *output;
  LADSPA_Data *input;

  LADSPA_Handle ladspa_handle;

  AgsFxDssiAudioInputData* input_data[AGS_SEQUENCER_MAX_MIDI_KEYS];
};
```

struct AgsFxDssiAudioInputData

```
struct AgsFxDssiAudioInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  LADSPA_Data *output;
  LADSPA_Data *input;

  LADSPA_Handle ladspa_handle;

  snd_seq_event_t *event_buffer;
  guint key_on;
};
```

AGS_TYPE_FX_DSSI_AUDIO

```
#define AGS_TYPE_FX_DSSI_AUDIO                (ags_fx_dssi_audio_get_type())
```

struct AgsFxDssiAudio

```
struct AgsFxDssiAudio;
```

struct AgsFxDssiAudioClass

```
struct AgsFxDssiAudioClass {
  AgsFxNotationAudioClass fx_notation_audio;
};
```

20.14 AgsFxDssiAudioProcessor

AgsFxDssiAudioProcessor — fx dssi audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxDssiAudioProcessor *	ags_fx_dssi_audio_processor_new ()
#define	AGS_FX_DSSI_AUDIO_PROCESSOR()
#define	AGS_FX_DSSI_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_DSSI_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_DSSI_AUDIO_PROCESSOR()
#define	AGS_IS_FX_DSSI_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_dssi_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR
struct	AgsFxDssiAudioProcessor
struct	AgsFxDssiAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioProcessor
                &#x2570;&#x2500;&#x2500; AgsFxDssiAudioProcessor
```

Implemented Interfaces

AgsFxDssiAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_dssi_audio_processor.h>
```

Description

The AgsFxDssiAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_dssi_audio_processor_new ()

```
AgsFxDssiAudioProcessor~*
ags_fx_dssi_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxDssiAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxDssiAudioProcessor

Since: 3.3.0

AGS_FX_DSSI_AUDIO_PROCESSOR()

```
#define AGS_FX_DSSI_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR, AgsFxDssiAudioProcessor))
```

AGS_FX_DSSI_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_DSSI_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR, AgsFxDssiAudioProcessorClass))
```

AGS_FX_DSSI_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_DSSI_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR, AgsFxDssiAudioProcessorClass))
```

AGS_IS_FX_DSSI_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_DSSI_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
     AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR))
```

AGS_IS_FX_DSSI_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_DSSI_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR))
```

ags_fx_dssi_audio_processor_get_type ()

```
GType
ags_fx_dssi_audio_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR                ( ↩
    ags_fx_dssi_audio_processor_get_type())
```

struct AgsFxDssiAudioProcessor

```
struct AgsFxDssiAudioProcessor;
```

struct AgsFxDssiAudioProcessorClass

```
struct AgsFxDssiAudioProcessorClass {
  AgsFxNotationAudioProcessorClass fx_notation_audio_processor;
};
```

20.15 AgsFxDssiAudioSignal

AgsFxDssiAudioSignal — fx dssi audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxDssiAudioSignal *	ags_fx_dssi_audio_signal_new ()
#define	AGS_FX_DSSI_AUDIO_SIGNAL()
#define	AGS_FX_DSSI_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_DSSI_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_DSSI_AUDIO_SIGNAL()
#define	AGS_IS_FX_DSSI_AUDIO_SIGNAL_CLASS()
GType	ags_fx_dssi_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_DSSI_AUDIO_SIGNAL
struct	AgsFxDssiAudioSignal
struct	AgsFxDssiAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioSignal
                &#x2570;&#x2500;&#x2500; AgsFxDssiAudioSignal
```

Implemented Interfaces

AgsFxDssiAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_dssi_audio_signal.h>
```

Description

The AgsFxDssiAudioSignal class provides ports to the effect processor.

Functions

ags_fx_dssi_audio_signal_new ()

```
AgsFxDssiAudioSignal~*
ags_fx_dssi_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxDssiAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxDssiAudioSignal

Since: 3.3.0

AGS_FX_DSSI_AUDIO_SIGNAL()

```
#define AGS_FX_DSSI_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_SIGNAL, AgsFxDssiAudioSignal))
```

AGS_FX_DSSI_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_DSSI_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_SIGNAL, AgsFxDssiAudioSignalClass))
```

AGS_FX_DSSI_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_DSSI_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_SIGNAL, AgsFxDssiAudioSignalClass))
```

AGS_IS_FX_DSSI_AUDIO_SIGNAL()

```
#define AGS_IS_FX_DSSI_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_SIGNAL))
```

AGS_IS_FX_DSSI_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_DSSI_AUDIO_SIGNAL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_DSSI_AUDIO_SIGNAL))
```

ags_fx_dssi_audio_signal_get_type ()

```
GType
ags_fx_dssi_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_DSSI_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_DSSI_AUDIO_SIGNAL                 (ags_fx_dssi_audio_signal_get_type())
```

struct AgsFxDssiAudioSignal

```
struct AgsFxDssiAudioSignal;
```

struct AgsFxDssiAudioSignalClass

```
struct AgsFxDssiAudioSignalClass {
  AgsFxNotationAudioSignalClass fx_notation_audio_signal;
};
```

20.16 AgsFxDssiChannel

AgsFxDssiChannel — fx dssi channel

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_fx_dssi_channel_load_port ()
AgsFxDssiChannel *	ags_fx_dssi_channel_new ()
#define	AGS_FX_DSSI_CHANNEL()
#define	AGS_FX_DSSI_CHANNEL_CLASS()
#define	AGS_FX_DSSI_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_DSSI_CHANNEL()
#define	AGS_IS_FX_DSSI_CHANNEL_CLASS()
GType	ags_fx_dssi_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_DSSI_CHANNEL
struct	AgsFxDssiChannel
struct	AgsFxDssiChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannel
                &#x2570;&#x2500;&#x2500; AgsFxDssiChannel
```

Implemented Interfaces

AgsFxDssiChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_dssi_channel.h>
```

Description

The AgsFxDssiChannel class provides ports to the effect processor.

Functions

ags_fx_dssi_channel_load_port ()

```
void
ags_fx_dssi_channel_load_port (AgsFxDssiChannel *fx_dssi_channel);
```

Load port of *fx_dssi_channel* .

Parameters

fx_dssi_channel	the AgsFxDssiChannel

Since: 3.3.0

ags_fx_dssi_channel_new ()

```
AgsFxDssiChannel~*
ags_fx_dssi_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxDssiChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxDssiChannel

Since: 3.3.0

AGS_FX_DSSI_CHANNEL()

```
#define AGS_FX_DSSI_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_DSSI_CHANNEL, AgsFxDssiChannel))
```

AGS_FX_DSSI_CHANNEL_CLASS()

```
#define AGS_FX_DSSI_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_DSSI_CHANNEL, AgsFxDssiChannelClass))
```

AGS_FX_DSSI_CHANNEL_GET_CLASS()

```
#define AGS_FX_DSSI_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_DSSI_CHANNEL, AgsFxDssiChannelClass))
```

AGS_IS_FX_DSSI_CHANNEL()

```
#define AGS_IS_FX_DSSI_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_DSSI_CHANNEL))
```

AGS_IS_FX_DSSI_CHANNEL_CLASS()

```
#define AGS_IS_FX_DSSI_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_DSSI_CHANNEL))
```

ags_fx_dssi_channel_get_type ()

```
GType
ags_fx_dssi_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_DSSI_CHANNEL

```
#define AGS_TYPE_FX_DSSI_CHANNEL                (ags_fx_dssi_channel_get_type())
```

struct AgsFxDssiChannel

```
struct AgsFxDssiChannel;
```

struct AgsFxDssiChannelClass

```
struct AgsFxDssiChannelClass {
  AgsFxNotationChannelClass fx_notation_channel;
};
```

20.17 AgsFxDssiChannelProcessor

AgsFxDssiChannelProcessor — fx dssi channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxDssiChannelProcessor *	ags_fx_dssi_channel_processor_new ()
#define	AGS_FX_DSSI_CHANNEL_PROCESSOR()
#define	AGS_FX_DSSI_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_DSSI_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_DSSI_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_DSSI_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_dssi_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR
struct	AgsFxDssiChannelProcessor
struct	AgsFxDssiChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannelProcessor
                &#x2570;&#x2500;&#x2500; AgsFxDssiChannelProcessor
```

Implemented Interfaces

AgsFxDssiChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_dssi_channel_processor.h>
```

Description

The AgsFxDssiChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_dssi_channel_processor_new ()

```
AgsFxDssiChannelProcessor~*
ags_fx_dssi_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxDssiChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxDssiChannelProcessor

Since: 3.3.0

AGS_FX_DSSI_CHANNEL_PROCESSOR()

```
#define AGS_FX_DSSI_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR, AgsFxDssiChannelProcessor))
```

AGS_FX_DSSI_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_DSSI_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR, AgsFxDssiChannelProcessorClass))
```

AGS_FX_DSSI_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_DSSI_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR, AgsFxDssiChannelProcessorClass))
```

AGS_IS_FX_DSSI_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_DSSI_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR))
```

AGS_IS_FX_DSSI_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_DSSI_CHANNEL_PROCESSOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR))
```

ags_fx_dssi_channel_processor_get_type ()

```
GType
ags_fx_dssi_channel_processor_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR                ( ↩
    ags_fx_dssi_channel_processor_get_type())
```

struct AgsFxDssiChannelProcessor

```
struct AgsFxDssiChannelProcessor;
```

struct AgsFxDssiChannelProcessorClass

```
struct AgsFxDssiChannelProcessorClass {
  AgsFxNotationChannelProcessorClass fx_notation_channel_processor;
};
```

20.18 AgsFxDssiRecycling

AgsFxDssiRecycling — fx dssi recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxDssiRecycling *	ags_fx_dssi_recycling_new ()
#define	AGS_FX_DSSI_RECYCLING()
#define	AGS_FX_DSSI_RECYCLING_CLASS()
#define	AGS_FX_DSSI_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_DSSI_RECYCLING()
#define	AGS_IS_FX_DSSI_RECYCLING_CLASS()
GType	ags_fx_dssi_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_DSSI_RECYCLING
struct	AgsFxDssiRecycling
struct	AgsFxDssiRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxNotationRecycling
                &#x2570;&#x2500;&#x2500; AgsFxDssiRecycling
```

Implemented Interfaces

AgsFxDssiRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_dssi_recycling.h>
```

Description

The AgsFxDssiRecycling class provides ports to the effect processor.

Functions

ags_fx_dssi_recycling_new ()

```
AgsFxDssiRecycling~*
ags_fx_dssi_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxDssiRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxDssiRecycling

Since: 3.3.0

AGS_FX_DSSI_RECYCLING()

```
#define AGS_FX_DSSI_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_DSSI_RECYCLING, AgsFxDssiRecycling))
```

AGS_FX_DSSI_RECYCLING_CLASS()

```
#define AGS_FX_DSSI_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_DSSI_RECYCLING, AgsFxDssiRecyclingClass))
```

AGS_FX_DSSI_RECYCLING_GET_CLASS()

```
#define AGS_FX_DSSI_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_DSSI_RECYCLING, AgsFxDssiRecyclingClass))
```

AGS_IS_FX_DSSI_RECYCLING()

```
#define AGS_IS_FX_DSSI_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_DSSI_RECYCLING))
```

AGS_IS_FX_DSSI_RECYCLING_CLASS()

```
#define AGS_IS_FX_DSSI_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_DSSI_RECYCLING))
```

ags_fx_dssi_recycling_get_type ()

```
GType
ags_fx_dssi_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_DSSI_RECYCLING

```
#define AGS_TYPE_FX_DSSI_RECYCLING                (ags_fx_dssi_recycling_get_type())
```

struct AgsFxDssiRecycling

```
struct AgsFxDssiRecycling;
```

struct AgsFxDssiRecyclingClass

```
struct AgsFxDssiRecyclingClass {
  AgsFxNotationRecyclingClass fx_notation_recycling;
};
```

20.19 AgsFxLv2Audio

AgsFxLv2Audio — fx lv2 audio

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_LV2_AUDIO_SCOPE_DATA()
#define	AGS_FX_LV2_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX()
#define	AGS_FX_LV2_AUDIO_CHANNEL_DATA()
#define	AGS_FX_LV2_AUDIO_CHANNEL_DATA_GET_STRCT_MUTEX()
#define	AGS_FX_LV2_AUDIO_INPUT_DATA()
#define	AGS_FX_LV2_AUDIO_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxLv2AudioScopeData *	ags_fx_lv2_audio_scope_data_alloc ()
void	ags_fx_lv2_audio_scope_data_free ()
AgsFxLv2AudioChannelData *	ags_fx_lv2_audio_channel_data_alloc ()
void	ags_fx_lv2_audio_channel_data_free ()
AgsFxLv2AudioInputData *	ags_fx_lv2_audio_input_data_alloc ()
void	ags_fx_lv2_audio_input_data_free ()
gboolean	ags_fx_lv2_audio_test_flags ()
void	ags_fx_lv2_audio_set_flags ()
void	ags_fx_lv2_audio_unset_flags ()
void	ags_fx_lv2_audio_load_plugin ()
void	ags_fx_lv2_audio_load_port ()
void	ags_fx_lv2_audio_change_program ()
AgsFxLv2Audio *	ags_fx_lv2_audio_new ()
#define	AGS_FX_LV2_AUDIO()
#define	AGS_FX_LV2_AUDIO_CLASS()
#define	AGS_FX_LV2_AUDIO_GET_CLASS()
#define	AGS_IS_FX_LV2_AUDIO()
#define	AGS_IS_FX_LV2_AUDIO_CLASS()
GType	ags_fx_lv2_audio_get_type ()

Types and Values

#define	AGS_FX_LV2_AUDIO_DEFAULT_MIDI_LENGHT
enum	AgsFxLv2AudioFlags
struct	AgsFxLv2AudioScopeData
struct	AgsFxLv2AudioChannelData
struct	AgsFxLv2AudioInputData
#define	AGS_TYPE_FX_LV2_AUDIO
struct	AgsFxLv2Audio
struct	AgsFxLv2AudioClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudio
        &#x2570;&#x2500;&#x2500; AgsFxNotationAudio
            &#x2570;&#x2500;&#x2500; AgsFxLv2Audio
```

Implemented Interfaces

AgsFxLv2Audio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lv2_audio.h>
```

Description

The AgsFxLv2Audio class provides ports to the effect processor.

Functions

AGS_FX_LV2_AUDIO_SCOPE_DATA()

```
#define AGS_FX_LV2_AUDIO_SCOPE_DATA(ptr) ((AgsFxLv2AudioScopeData *) (ptr))
```

AGS_FX_LV2_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_LV2_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxLv2AudioScopeData *)(ptr ↩
    ))->strct_mutex))
```

AGS_FX_LV2_AUDIO_CHANNEL_DATA()

```
#define AGS_FX_LV2_AUDIO_CHANNEL_DATA(ptr) ((AgsFxLv2AudioChannelData *) (ptr))
```

AGS_FX_LV2_AUDIO_CHANNEL_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_LV2_AUDIO_CHANNEL_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxLv2AudioChannelData *) ↩
    (ptr))->strct_mutex))
```

AGS_FX_LV2_AUDIO_INPUT_DATA()

```
#define AGS_FX_LV2_AUDIO_INPUT_DATA(ptr) ((AgsFxLv2AudioInputData *) (ptr))
```

AGS_FX_LV2_AUDIO_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_LV2_AUDIO_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxLv2AudioInputData *)(ptr ↩
    ))->strct_mutex))
```

ags_fx_lv2_audio_scope_data_alloc ()

```
AgsFxLv2AudioScopeData~*
ags_fx_lv2_audio_scope_data_alloc ();
```

Allocate AgsFxLv2AudioScopeData

Returns

the new AgsFxLv2AudioScopeData

Since: 3.3.0

ags_fx_lv2_audio_scope_data_free ()

```
void
ags_fx_lv2_audio_scope_data_free (AgsFxLv2AudioScopeData *scope_data);
```

Free *scope_data* .

Parameters

scope_data	the AgsFxLv2AudioScopeData

Since: 3.3.0

ags_fx_lv2_audio_channel_data_alloc ()

```
AgsFxLv2AudioChannelData~*
ags_fx_lv2_audio_channel_data_alloc ();
```

Allocate AgsFxLv2AudioChannelData

Returns

the new AgsFxLv2AudioChannelData

Since: 3.3.0

ags_fx_lv2_audio_channel_data_free ()

```
void
ags_fx_lv2_audio_channel_data_free (AgsFxLv2AudioChannelData *channel_data);
```

Free *channel_data* .

Parameters

channel_data	the AgsFxLv2AudioChannelData

Since: 3.3.0

ags_fx_lv2_audio_input_data_alloc ()

```
AgsFxLv2AudioInputData~*
ags_fx_lv2_audio_input_data_alloc ();
```

Allocate AgsFxLv2AudioInputData

Returns

the new AgsFxLv2AudioInputData

Since: 3.3.0

ags_fx_lv2_audio_input_data_free ()

```
void
ags_fx_lv2_audio_input_data_free (AgsFxLv2AudioInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxLv2AudioInputData	

Since: 3.3.0

ags_fx_lv2_audio_test_flags ()

```
gboolean
ags_fx_lv2_audio_test_flags (AgsFxLv2Audio *fx_lv2_audio,
                             guint flags);
```

Test *flags* of *fx_lv2_audio* .

Parameters

fx_lv2_audio	the AgsFxLv2Audio	
flags	the flags	

Returns

TRUE on success, otherwise FALSE

Since: 3.3.0

ags_fx_lv2_audio_set_flags ()

```
void
ags_fx_lv2_audio_set_flags (AgsFxLv2Audio *fx_lv2_audio,
                            guint flags);
```

Set *flags* of *fx_lv2_audio* .

Parameters

fx_lv2_audio	the AgsFxLv2Audio	
flags	the flags	

Since: 3.3.0

ags_fx_lv2_audio_unset_flags ()

```
void
ags_fx_lv2_audio_unset_flags (AgsFxLv2Audio *fx_lv2_audio,
                              guint flags);
```

Unset *flags* of *fx_lv2_audio* .

Parameters

fx_lv2_audio	the AgsFxLv2Audio	
flags	the flags	

Since: 3.3.0

ags_fx_lv2_audio_load_plugin ()

```
void
ags_fx_lv2_audio_load_plugin (AgsFxLv2Audio *fx_lv2_audio);
```

Load plugin of *fx_lv2_audio* .

Parameters

fx_lv2_audio	the AgsFxLv2Audio	

Since: 3.3.0

ags_fx_lv2_audio_load_port ()

```
void
ags_fx_lv2_audio_load_port (AgsFxLv2Audio *fx_lv2_audio);
```

ags_fx_lv2_audio_change_program ()

```
void
ags_fx_lv2_audio_change_program (AgsFxLv2Audio *fx_lv2_audio,
                                 guint bank_index,
                                 guint program_index);
```

Change program of *fx_lv2_audio* .

Parameters

fx_lv2_audio	the AgsFxLv2Audio	
bank_index	the bank index	
program_index	the program index	

Since: 3.3.0

ags_fx_lv2_audio_new ()

```
AgsFxLv2Audio~*
ags_fx_lv2_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxLv2Audio

Parameters

audio	the AgsAudio

Returns

the new AgsFxLv2Audio

Since: 3.3.0

AGS_FX_LV2_AUDIO()

```
#define AGS_FX_LV2_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO, AgsFxLv2Audio))
```

AGS_FX_LV2_AUDIO_CLASS()

```
#define AGS_FX_LV2_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LV2_AUDIO, AgsFxLv2AudioClass))
```

AGS_FX_LV2_AUDIO_GET_CLASS()

```
#define AGS_FX_LV2_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO, AgsFxLv2AudioClass))
```

AGS_IS_FX_LV2_AUDIO()

```
#define AGS_IS_FX_LV2_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO))
```

AGS_IS_FX_LV2_AUDIO_CLASS()

```
#define AGS_IS_FX_LV2_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LV2_AUDIO))
```

ags_fx_lv2_audio_get_type ()

```
GType
ags_fx_lv2_audio_get_type ();
```

Types and Values

AGS_FX_LV2_AUDIO_DEFAULT_MIDI_LENGHT

```
#define AGS_FX_LV2_AUDIO_DEFAULT_MIDI_LENGHT (8 * 256)
```

enum AgsFxLv2AudioFlags

Members

AGS_FX_LV2_AUDIO_LIVE_INSTRUMENT	

struct AgsFxLv2AudioScopeData

```
struct AgsFxLv2AudioScopeData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint audio_channels;

  AgsFxLv2AudioChannelData **channel_data;
};
```

struct AgsFxLv2AudioChannelData

```
struct AgsFxLv2AudioChannelData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint event_count;

  float *output;
  float *input;

  gpointer event_port;
  gpointer atom_port;

  LV2_Handle *lv2_handle;

  AgsFxLv2AudioInputData* input_data[AGS_SEQUENCER_MAX_MIDI_KEYS];
};
```

struct AgsFxLv2AudioInputData

```
struct AgsFxLv2AudioInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  float *output;
  float *input;
```

```
  gpointer event_port;
  gpointer atom_port;

  LV2_Handle *lv2_handle;

  snd_seq_event_t *event_buffer;
  guint key_on;
};
```

AGS_TYPE_FX_LV2_AUDIO

```
#define AGS_TYPE_FX_LV2_AUDIO                (ags_fx_lv2_audio_get_type())
```

struct AgsFxLv2Audio

```
struct AgsFxLv2Audio;
```

struct AgsFxLv2AudioClass

```
struct AgsFxLv2AudioClass {
  AgsFxNotationAudioClass fx_notation_audio;
};
```

20.20 AgsFxLv2AudioProcessor

AgsFxLv2AudioProcessor — fx lv2 audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLv2AudioProcessor *	ags_fx_lv2_audio_processor_new ()
#define	AGS_FX_LV2_AUDIO_PROCESSOR()
#define	AGS_FX_LV2_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_LV2_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LV2_AUDIO_PROCESSOR()
#define	AGS_IS_FX_LV2_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_lv2_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LV2_AUDIO_PROCESSOR
struct	AgsFxLv2AudioProcessor
struct	AgsFxLv2AudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioProcessor
                &#x2570;&#x2500;&#x2500; AgsFxLv2AudioProcessor
```

Implemented Interfaces

AgsFxLv2AudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_lv2_audio_processor.h>
```

Description

The AgsFxLv2AudioProcessor class provides ports to the effect processor.

Functions

ags_fx_lv2_audio_processor_new ()

```
AgsFxLv2AudioProcessor~*
ags_fx_lv2_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxLv2AudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxLv2AudioProcessor

Since: 3.3.0

AGS_FX_LV2_AUDIO_PROCESSOR()

```
#define AGS_FX_LV2_AUDIO_PROCESSOR(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO_PROCESSOR, AgsFxLv2AudioProcessor))
```

AGS_FX_LV2_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_LV2_AUDIO_PROCESSOR_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LV2_AUDIO_PROCESSOR, AgsFxLv2AudioProcessorClass))
```

AGS_FX_LV2_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LV2_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO_PROCESSOR, AgsFxLv2AudioProcessorClass))
```

AGS_IS_FX_LV2_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_LV2_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO_PROCESSOR))
```

AGS_IS_FX_LV2_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LV2_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LV2_AUDIO_PROCESSOR))
```

ags_fx_lv2_audio_processor_get_type ()

```
GType
ags_fx_lv2_audio_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_LV2_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_LV2_AUDIO_PROCESSOR                (ags_fx_lv2_audio_processor_get_type ↩
    ())
```

struct AgsFxLv2AudioProcessor

```
struct AgsFxLv2AudioProcessor;
```

struct AgsFxLv2AudioProcessorClass

```
struct AgsFxLv2AudioProcessorClass {
  AgsFxNotationAudioProcessorClass fx_notation_audio_processor;
};
```

20.21 AgsFxLv2AudioSignal

AgsFxLv2AudioSignal — fx lv2 audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLv2AudioSignal *	ags_fx_lv2_audio_signal_new ()
#define	AGS_FX_LV2_AUDIO_SIGNAL()
#define	AGS_FX_LV2_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_LV2_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_LV2_AUDIO_SIGNAL()
#define	AGS_IS_FX_LV2_AUDIO_SIGNAL_CLASS()
GType	ags_fx_lv2_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_LV2_AUDIO_SIGNAL
struct	AgsFxLv2AudioSignal
struct	AgsFxLv2AudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioSignal
                &#x2570;&#x2500;&#x2500; AgsFxLv2AudioSignal
```

Implemented Interfaces

AgsFxLv2AudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lv2_audio_signal.h>
```

Description

The AgsFxLv2AudioSignal class provides ports to the effect processor.

Functions

ags_fx_lv2_audio_signal_new ()

```
AgsFxLv2AudioSignal~*
ags_fx_lv2_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxLv2AudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxLv2AudioSignal

Since: 3.3.0

AGS_FX_LV2_AUDIO_SIGNAL()

```
#define AGS_FX_LV2_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO_SIGNAL, AgsFxLv2AudioSignal))
```

AGS_FX_LV2_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_LV2_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LV2_AUDIO_SIGNAL, AgsFxLv2AudioSignalClass))
```

AGS_FX_LV2_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_LV2_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO_SIGNAL, AgsFxLv2AudioSignalClass))
```

AGS_IS_FX_LV2_AUDIO_SIGNAL()

```
#define AGS_IS_FX_LV2_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LV2_AUDIO_SIGNAL))
```

AGS_IS_FX_LV2_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_LV2_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LV2_AUDIO_SIGNAL))
```

ags_fx_lv2_audio_signal_get_type ()

```
GType
ags_fx_lv2_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_LV2_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_LV2_AUDIO_SIGNAL                (ags_fx_lv2_audio_signal_get_type())
```

struct AgsFxLv2AudioSignal

```
struct AgsFxLv2AudioSignal;
```

struct AgsFxLv2AudioSignalClass

```
struct AgsFxLv2AudioSignalClass {
  AgsFxNotationAudioSignalClass fx_notation_audio_signal;
};
```

20.22 AgsFxLv2Channel

AgsFxLv2Channel — fx lv2 channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_LV2_CHANNEL_INPUT_DATA()
#define	AGS_FX_LV2_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxLv2ChannelInputData *	ags_fx_lv2_channel_input_data_alloc ()
void	ags_fx_lv2_channel_input_data_free ()
void	ags_fx_lv2_channel_load_plugin ()
void	ags_fx_lv2_channel_load_port ()
AgsFxLv2Channel *	ags_fx_lv2_channel_new ()
#define	AGS_FX_LV2_CHANNEL()
#define	AGS_FX_LV2_CHANNEL_CLASS()
#define	AGS_FX_LV2_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_LV2_CHANNEL()
#define	AGS_IS_FX_LV2_CHANNEL_CLASS()
GType	ags_fx_lv2_channel_get_type ()

Types and Values

struct	AgsFxLv2ChannelInputData
#define	AGS_TYPE_FX_LV2_CHANNEL
struct	AgsFxLv2Channel
struct	AgsFxLv2ChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannel
                &#x2570;&#x2500;&#x2500; AgsFxLv2Channel
```

Implemented Interfaces

AgsFxLv2Channel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lv2_channel.h>
```

Description

The AgsFxLv2Channel class provides ports to the effect processor.

Functions

AGS_FX_LV2_CHANNEL_INPUT_DATA()

```
#define AGS_FX_LV2_CHANNEL_INPUT_DATA(ptr) ((AgsFxLv2ChannelInputData *)(ptr))
```

AGS_FX_LV2_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_LV2_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxLv2ChannelInputData *) ↩
    (ptr))->strct_mutex))
```

ags_fx_lv2_channel_input_data_alloc ()

```
AgsFxLv2ChannelInputData~*
ags_fx_lv2_channel_input_data_alloc ();
```

Allocate AgsFxLv2ChannelInputData

Returns

the new AgsFxLv2ChannelInputData

Since: 3.3.0

ags_fx_lv2_channel_input_data_free ()

```
void
ags_fx_lv2_channel_input_data_free (AgsFxLv2ChannelInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxLv2ChannelInputData

Since: 3.3.0

ags_fx_lv2_channel_load_plugin ()

```
void
ags_fx_lv2_channel_load_plugin (AgsFxLv2Channel *fx_lv2_channel);
```

Load plugin of *fx_lv2_channel* .

Parameters

fx_lv2_channel	the AgsFxLv2Channel

Since: 3.3.0

ags_fx_lv2_channel_load_port ()

```
void
ags_fx_lv2_channel_load_port (AgsFxLv2Channel *fx_lv2_channel);
```

Load port of *fx_lv2_channel* .

Parameters

fx_lv2_channel	the AgsFxLv2Channel

Since: 3.3.0

ags_fx_lv2_channel_new ()

```
AgsFxLv2Channel~*
ags_fx_lv2_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxLv2Channel

Parameters

channel	the AgsChannel

Returns

the new AgsFxLv2Channel

Since: 3.3.0

AGS_FX_LV2_CHANNEL()

```
#define AGS_FX_LV2_CHANNEL(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LV2_CHANNEL, AgsFxLv2Channel))
```

AGS_FX_LV2_CHANNEL_CLASS()

```
#define AGS_FX_LV2_CHANNEL_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LV2_CHANNEL, AgsFxLv2ChannelClass))
```

AGS_FX_LV2_CHANNEL_GET_CLASS()

```
#define AGS_FX_LV2_CHANNEL_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LV2_CHANNEL, AgsFxLv2ChannelClass))
```

AGS_IS_FX_LV2_CHANNEL()

```
#define AGS_IS_FX_LV2_CHANNEL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LV2_CHANNEL))
```

AGS_IS_FX_LV2_CHANNEL_CLASS()

```
#define AGS_IS_FX_LV2_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LV2_CHANNEL))
```

ags_fx_lv2_channel_get_type ()

```
GType
ags_fx_lv2_channel_get_type ();
```

Types and Values

struct AgsFxLv2ChannelInputData

```
struct AgsFxLv2ChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  float *output;
  float *input;

  LV2_Handle *lv2_handle;
};
```

AGS_TYPE_FX_LV2_CHANNEL

```
#define AGS_TYPE_FX_LV2_CHANNEL                 (ags_fx_lv2_channel_get_type())
```

struct AgsFxLv2Channel

```
struct AgsFxLv2Channel;
```

struct AgsFxLv2ChannelClass

```
struct AgsFxLv2ChannelClass {
  AgsFxNotationChannelClass fx_notation_channel;
};
```

20.23 AgsFxLv2ChannelProcessor

AgsFxLv2ChannelProcessor — fx lv2 channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLv2ChannelProcessor *	ags_fx_lv2_channel_processor_new ()
#define	AGS_FX_LV2_CHANNEL_PROCESSOR()
#define	AGS_FX_LV2_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_LV2_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LV2_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_LV2_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_lv2_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR
struct	AgsFxLv2ChannelProcessor
struct	AgsFxLv2ChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannelProcessor
                &#x2570;&#x2500;&#x2500; AgsFxLv2ChannelProcessor
```

Implemented Interfaces

AgsFxLv2ChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lv2_channel_processor.h>
```

Description

The AgsFxLv2ChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_lv2_channel_processor_new ()

```
AgsFxLv2ChannelProcessor~*
ags_fx_lv2_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxLv2ChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxLv2ChannelProcessor

Since: 3.3.0

AGS_FX_LV2_CHANNEL_PROCESSOR()

```
#define AGS_FX_LV2_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR, AgsFxLv2ChannelProcessor))
```

AGS_FX_LV2_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_LV2_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR, AgsFxLv2ChannelProcessorClass))
```

AGS_FX_LV2_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LV2_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR, AgsFxLv2ChannelProcessorClass))
```

AGS_IS_FX_LV2_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_LV2_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR))
```

AGS_IS_FX_LV2_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LV2_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR))
```

ags_fx_lv2_channel_processor_get_type ()

```
GType
ags_fx_lv2_channel_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR                ( ↩
    ags_fx_lv2_channel_processor_get_type())
```

struct AgsFxLv2ChannelProcessor

```
struct AgsFxLv2ChannelProcessor;
```

struct AgsFxLv2ChannelProcessorClass

```
struct AgsFxLv2ChannelProcessorClass {
  AgsFxNotationChannelProcessorClass fx_notation_channel_processor;
};
```

20.24 AgsFxLv2Recycling

AgsFxLv2Recycling — fx lv2 recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLv2Recycling *	ags_fx_lv2_recycling_new ()
#define	AGS_FX_LV2_RECYCLING()
#define	AGS_FX_LV2_RECYCLING_CLASS()
#define	AGS_FX_LV2_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_LV2_RECYCLING()
#define	AGS_IS_FX_LV2_RECYCLING_CLASS()
GType	ags_fx_lv2_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_LV2_RECYCLING
struct	AgsFxLv2Recycling
struct	AgsFxLv2RecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxNotationRecycling
                &#x2570;&#x2500;&#x2500; AgsFxLv2Recycling
```

Implemented Interfaces

AgsFxLv2Recycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lv2_recycling.h>
```

Description

The AgsFxLv2Recycling class provides ports to the effect processor.

Functions

ags_fx_lv2_recycling_new ()

```
AgsFxLv2Recycling~*
ags_fx_lv2_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxLv2Recycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxLv2Recycling

Since: 3.3.0

AGS_FX_LV2_RECYCLING()

```
#define AGS_FX_LV2_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LV2_RECYCLING, AgsFxLv2Recycling))
```

AGS_FX_LV2_RECYCLING_CLASS()

```
#define AGS_FX_LV2_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LV2_RECYCLING, AgsFxLv2RecyclingClass))
```

AGS_FX_LV2_RECYCLING_GET_CLASS()

```
#define AGS_FX_LV2_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LV2_RECYCLING, AgsFxLv2RecyclingClass))
```

AGS_IS_FX_LV2_RECYCLING()

```
#define AGS_IS_FX_LV2_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LV2_RECYCLING))
```

AGS_IS_FX_LV2_RECYCLING_CLASS()

```
#define AGS_IS_FX_LV2_RECYCLING_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LV2_RECYCLING))
```

ags_fx_lv2_recycling_get_type ()

```
GType
ags_fx_lv2_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_LV2_RECYCLING

```
#define AGS_TYPE_FX_LV2_RECYCLING                 (ags_fx_lv2_recycling_get_type())
```

struct AgsFxLv2Recycling

```
struct AgsFxLv2Recycling;
```

struct AgsFxLv2RecyclingClass

```
struct AgsFxLv2RecyclingClass {
  AgsFxNotationRecyclingClass fx_notation_recycling;
};
```

Synths extending ags-fx-notation.

20.25 AgsFxSynthAudio

AgsFxSynthAudio — fx synth audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSynthAudioScopeData *	ags_fx_synth_audio_scope_data_alloc ()
void	ags_fx_synth_audio_scope_data_free ()
AgsFxSynthAudioChannelData *	ags_fx_synth_audio_channel_data_alloc ()
void	ags_fx_synth_audio_channel_data_free ()
AgsFxSynthAudioInputData *	ags_fx_synth_audio_input_data_alloc ()
void	ags_fx_synth_audio_input_data_free ()
AgsFxSynthAudio *	ags_fx_synth_audio_new ()
#define	AGS_FX_SYNTH_AUDIO()

#define	AGS_FX_SYNTH_AUDIO_CLASS()
#define	AGS_FX_SYNTH_AUDIO_GET_CLASS()
#define	AGS_IS_FX_SYNTH_AUDIO()
#define	AGS_IS_FX_SYNTH_AUDIO_CLASS()
GType	ags_fx_synth_audio_get_type ()

Properties

AgsPort *	chorus-delay	Read / Write
AgsPort *	chorus-depth	Read / Write
AgsPort *	chorus-enabled	Read / Write
AgsPort *	chorus-input-volume	Read / Write
AgsPort *	chorus-lfo-frequency	Read / Write
AgsPort *	chorus-lfo-oscillator	Read / Write
AgsPort *	chorus-mix	Read / Write
AgsPort *	chorus-output-volume	Read / Write
AgsPort *	chorus-pitch-type	Read / Write
AgsPort *	high-pass-enabled	Read / Write
AgsPort *	high-pass-filter-gain	Read / Write
AgsPort *	high-pass-q-lin	Read / Write
AgsPort *	low-pass-enabled	Read / Write
AgsPort *	low-pass-filter-gain	Read / Write
AgsPort *	low-pass-q-lin	Read / Write
AgsPort *	noise-gain	Read / Write
AgsPort *	pitch-tuning	Read / Write
AgsPort *	pitch-type	Read / Write
AgsPort *	sequencer-enabled	Read / Write
AgsPort *	sequencer-sign	Read / Write
AgsPort *	synth-0-key	Read / Write
AgsPort *	synth-0-octave	Read / Write
AgsPort *	synth-0-oscillator	Read / Write
AgsPort *	synth-0-phase	Read / Write
AgsPort *	synth-0-sync-attack-0	Read / Write
AgsPort *	synth-0-sync-attack-1	Read / Write
AgsPort *	synth-0-sync-attack-2	Read / Write
AgsPort *	synth-0-sync-enabled	Read / Write
AgsPort *	synth-0-sync-lfo-frequency	Read / Write
AgsPort *	synth-0-sync-lfo-oscillator	Read / Write
AgsPort *	synth-0-sync-phase-0	Read / Write
AgsPort *	synth-0-sync-phase-1	Read / Write
AgsPort *	synth-0-sync-phase-2	Read / Write
AgsPort *	synth-0-sync-relative-attack-factor	Read / Write
AgsPort *	synth-0-volume	Read / Write
AgsPort *	synth-1-key	Read / Write
AgsPort *	synth-1-octave	Read / Write
AgsPort *	synth-1-oscillator	Read / Write
AgsPort *	synth-1-phase	Read / Write
AgsPort *	synth-1-sync-attack-0	Read / Write
AgsPort *	synth-1-sync-attack-1	Read / Write
AgsPort *	synth-1-sync-attack-2	Read / Write
AgsPort *	synth-1-sync-enabled	Read / Write
AgsPort *	synth-1-sync-lfo-frequency	Read / Write
AgsPort *	synth-1-sync-lfo-oscillator	Read / Write
AgsPort *	synth-1-sync-phase-0	Read / Write
AgsPort *	synth-1-sync-phase-1	Read / Write

AgsPort *	synth-1-sync-phase-2	Read / Write
AgsPort *	synth-1-sync-relative-attack-factor	Read / Write
AgsPort *	synth-1-volume	Read / Write

Types and Values

#define	AGS_FX_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE
struct	AgsFxSynthAudioScopeData
struct	AgsFxSynthAudioChannelData
struct	AgsFxSynthAudioInputData
#define	AGS_TYPE_FX_SYNTH_AUDIO
struct	AgsFxSynthAudio
struct	AgsFxSynthAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudio
                &#x2570;&#x2500;&#x2500; AgsFxSynthAudio
```

Implemented Interfaces

AgsFxSynthAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_synth_audio.h>
```

Description

The AgsFxSynthAudio class provides ports to the effect processor.

Functions

ags_fx_synth_audio_scope_data_alloc ()

```
AgsFxSynthAudioScopeData~*
ags_fx_synth_audio_scope_data_alloc ();
```

Allocate AgsFxSynthAudioScopeData

Returns

the new AgsFxSynthAudioScopeData

Since: 3.14.0

ags_fx_synth_audio_scope_data_free ()

```
void
ags_fx_synth_audio_scope_data_free (AgsFxSynthAudioScopeData *scope_data);
```

Free *scope_data* .

Parameters

scope_data	the AgsFxSynthAudioScopeData

Since: 3.14.0

ags_fx_synth_audio_channel_data_alloc ()

```
AgsFxSynthAudioChannelData~*
ags_fx_synth_audio_channel_data_alloc ();
```

Allocate AgsFxSynthAudioChannelData

Returns

the new AgsFxSynthAudioChannelData

Since: 3.14.0

ags_fx_synth_audio_channel_data_free ()

```
void
ags_fx_synth_audio_channel_data_free (AgsFxSynthAudioChannelData *channel_data);
```

Free *channel_data* .

Parameters

channel_data	the AgsFxSynthAudioChannelData

Since: 3.14.0

ags_fx_synth_audio_input_data_alloc ()

```
AgsFxSynthAudioInputData~*
ags_fx_synth_audio_input_data_alloc ();
```

Allocate AgsFxSynthAudioInputData

Returns

the new AgsFxSynthAudioInputData

Since: 3.14.0

ags_fx_synth_audio_input_data_free ()

```
void
ags_fx_synth_audio_input_data_free (AgsFxSynthAudioInputData *input_data);
```

Free *input_data*.

Parameters

input_data	the AgsFxSynthAudioInputData

Since: 3.14.0

ags_fx_synth_audio_new ()

```
AgsFxSynthAudio~*
ags_fx_synth_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxSynthAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxSynthAudio

Since: 3.14.0

AGS_FX_SYNTH_AUDIO()

```
#define AGS_FX_SYNTH_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO, AgsFxSynthAudio))
```

AGS_FX_SYNTH_AUDIO_CLASS()

```
#define AGS_FX_SYNTH_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO, AgsFxSynthAudioClass))
```

AGS_FX_SYNTH_AUDIO_GET_CLASS()

```
#define AGS_FX_SYNTH_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO, AgsFxSynthAudioClass))
```

AGS_IS_FX_SYNTH_AUDIO()

```
#define AGS_IS_FX_SYNTH_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO))
```

AGS_IS_FX_SYNTH_AUDIO_CLASS()

```
#define AGS_IS_FX_SYNTH_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO))
```

ags_fx_synth_audio_get_type ()

```
GType
ags_fx_synth_audio_get_type ();
```

Types and Values

AGS_FX_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE

```
#define AGS_FX_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE (65536)
```

struct AgsFxSynthAudioScopeData

```
struct AgsFxSynthAudioScopeData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint audio_channels;

  AgsFxSynthAudioChannelData **channel_data;
};
```

struct AgsFxSynthAudioChannelData

```
struct AgsFxSynthAudioChannelData {
  GRecMutex strct_mutex;

  gpointer parent;

  AgsSynthUtil *synth_0;
  AgsSynthUtil *synth_1;

  AgsNoiseUtil *noise_util;

  GType pitch_type;
  gpointer pitch_util;

  AgsChorusUtil *chorus_util;

  AgsFluidIIRFilterUtil *low_pass_filter;
  AgsFluidIIRFilterUtil *high_pass_filter;

  AgsFxSynthAudioInputData* input_data[AGS_SEQUENCER_MAX_MIDI_KEYS];
};
```

struct AgsFxSynthAudioInputData

```
struct AgsFxSynthAudioInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint key_on;
};
```

AGS_TYPE_FX_SYNTH_AUDIO

```
#define AGS_TYPE_FX_SYNTH_AUDIO                (ags_fx_synth_audio_get_type())
```

struct AgsFxSynthAudio

```
struct AgsFxSynthAudio;
```

struct AgsFxSynthAudioClass

```
struct AgsFxSynthAudioClass {
  AgsFxNotationAudioClass fx_notation_audio;
};
```

Property Details

The "chorus-delay" property

```
  "chorus-delay"             AgsPort~*
```

The chorus delay.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-depth" property

```
  "chorus-depth"             AgsPort~*
```

The chorus depth.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-enabled" property

```
"chorus-enabled"            AgsPort~*
```

The chorus enabled.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-input-volume" property

```
"chorus-input-volume"       AgsPort~*
```

The chorus input volume.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-lfo-frequency" property

```
"chorus-lfo-frequency"      AgsPort~*
```

The chorus lfo frequency.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-lfo-oscillator" property

```
"chorus-lfo-oscillator"     AgsPort~*
```

The chorus lfo oscillator.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-mix" property

```
"chorus-mix"                AgsPort~*
```

The chorus mix.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-output-volume" property

```
"chorus-output-volume"      AgsPort~*
```

The chorus output volume.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "chorus-pitch-type" property

```
"chorus-pitch-type"         AgsPort~*
```

The chorus pitch type.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 4.0.0

The "high-pass-enabled" property

```
"high-pass-enabled"         AgsPort~*
```

The high pass enabled.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "high-pass-filter-gain" property

```
"high-pass-filter-gain"     AgsPort~*
```

The high pass filter gain.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "high-pass-q-lin" property

```
"high-pass-q-lin"           AgsPort~*
```

The high pass q-lin.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "low-pass-enabled" property

```
"low-pass-enabled"          AgsPort~*
```

The low pass enabled.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "low-pass-filter-gain" property

```
"low-pass-filter-gain"      AgsPort~*
```

The low pass filter gain.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "low-pass-q-lin" property

```
"low-pass-q-lin"            AgsPort~*
```

The low pass q-lin.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "noise-gain" property

```
"noise-gain"                AgsPort~*
```

The noise gain.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "pitch-tuning" property

```
"pitch-tuning"              AgsPort~*
```

The pitch tuning.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "pitch-type" property

```
"pitch-type"                AgsPort~*
```

The pitch type.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 4.0.0

The "sequencer-enabled" property

```
"sequencer-enabled"         AgsPort~*
```

The sequencer enabled.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "sequencer-sign" property

```
"sequencer-sign"            AgsPort~*
```

The sequencer sign.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-key" property

```
"synth-0-key"               AgsPort~*
```

The synth-0 key.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-octave" property

```
"synth-0-octave"            AgsPort~*
```

The synth-0 octave.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-oscillator" property

```
"synth-0-oscillator"        AgsPort~*
```

The synth-0 oscillator.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-phase" property

```
"synth-0-phase"             AgsPort~*
```

The synth-0 phase.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-attack-0" property

```
"synth-0-sync-attack-0"     AgsPort~*
```

The synth-0 sync attack-0.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-attack-1" property

```
"synth-0-sync-attack-1"     AgsPort~*
```

The synth-0 sync attack-1.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-attack-2" property

```
"synth-0-sync-attack-2"     AgsPort~*
```

The synth-0 sync attack-2.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-enabled" property

```
"synth-0-sync-enabled"     AgsPort~*
```

The synth-0 sync enabled.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-lfo-frequency" property

```
"synth-0-sync-lfo-frequency" AgsPort~*
```

The synth-0 sync LFO frequency.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-lfo-oscillator" property

```
"synth-0-sync-lfo-oscillator" AgsPort~*
```

The synth-0 sync LFO oscillator.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-phase-0" property

```
"synth-0-sync-phase-0"     AgsPort~*
```

The synth-0 sync phase-0.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-phase-1" property

```
"synth-0-sync-phase-1"     AgsPort~*
```

The synth-0 sync phase-1.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-phase-2" property

```
"synth-0-sync-phase-2"      AgsPort~*
```

The synth-0 sync phase-2.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-sync-relative-attack-factor" property

```
"synth-0-sync-relative-attack-factor" AgsPort~*
```

The synth-0 sync relative attack factor.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-0-volume" property

```
"synth-0-volume"            AgsPort~*
```

The synth-0 volume.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-key" property

```
"synth-1-key"               AgsPort~*
```

The synth-1 key.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-octave" property

```
"synth-1-octave"            AgsPort~*
```

The synth-1 octave.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-oscillator" property

```
"synth-1-oscillator"        AgsPort~*
```

The synth-1 oscillator.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-phase" property

```
"synth-1-phase"             AgsPort~*
```

The synth-1 phase.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-attack-0" property

```
"synth-1-sync-attack-0"     AgsPort~*
```

The synth-1 sync attack-0.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-attack-1" property

```
"synth-1-sync-attack-1"     AgsPort~*
```

The synth-1 sync attack-1.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-attack-2" property

```
"synth-1-sync-attack-2"     AgsPort~*
```

The synth-1 sync attack-2.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-enabled" property

```
"synth-1-sync-enabled"     AgsPort~*
```

The synth-1 sync enabled.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-lfo-frequency" property

```
"synth-1-sync-lfo-frequency" AgsPort~*
```

The synth-1 sync LFO frequency.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-lfo-oscillator" property

```
"synth-1-sync-lfo-oscillator" AgsPort~*
```

The synth-1 sync LFO oscillator.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-phase-0" property

```
"synth-1-sync-phase-0"     AgsPort~*
```

The synth-1 sync phase-0.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-phase-1" property

```
"synth-1-sync-phase-1"     AgsPort~*
```

The synth-1 sync phase-1.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-phase-2" property

```
"synth-1-sync-phase-2"      AgsPort~*
```

The synth-1 sync phase-2.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-sync-relative-attack-factor" property

```
"synth-1-sync-relative-attack-factor" AgsPort~*
```

The synth-1 sync relative attack factor.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

The "synth-1-volume" property

```
"synth-1-volume"            AgsPort~*
```

The synth-1 volume.

Owner: AgsFxSynthAudio

Flags: Read / Write

Since: 3.14.0

20.26 AgsFxSynthAudioProcessor

AgsFxSynthAudioProcessor — fx synth audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSynthAudioProcessor *	ags_fx_synth_audio_processor_new ()
#define	AGS_FX_SYNTH_AUDIO_PROCESSOR()
#define	AGS_FX_SYNTH_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_SYNTH_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_SYNTH_AUDIO_PROCESSOR()
#define	AGS_IS_FX_SYNTH_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_synth_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR
struct	AgsFxSynthAudioProcessor
struct	AgsFxSynthAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioProcessor
                &#x2570;&#x2500;&#x2500; AgsFxSynthAudioProcessor
```

Implemented Interfaces

AgsFxSynthAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_synth_audio_processor.h>
```

Description

The AgsFxSynthAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_synth_audio_processor_new ()

```
AgsFxSynthAudioProcessor~*
ags_fx_synth_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxSynthAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxSynthAudioProcessor

Since: 3.14.0

AGS_FX_SYNTH_AUDIO_PROCESSOR()

```
#define AGS_FX_SYNTH_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR, AgsFxSynthAudioProcessor))
```

AGS_FX_SYNTH_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_SYNTH_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR, AgsFxSynthAudioProcessorClass))
```

AGS_FX_SYNTH_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_SYNTH_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR, AgsFxSynthAudioProcessorClass))
```

AGS_IS_FX_SYNTH_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_SYNTH_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR))
```

AGS_IS_FX_SYNTH_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_SYNTH_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR))
```

ags_fx_synth_audio_processor_get_type ()

```
GType
ags_fx_synth_audio_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR                ( ↩
    ags_fx_synth_audio_processor_get_type())
```

struct AgsFxSynthAudioProcessor

```
struct AgsFxSynthAudioProcessor;
```

struct AgsFxSynthAudioProcessorClass

```
struct AgsFxSynthAudioProcessorClass {
  AgsFxNotationAudioProcessorClass fx_notation_audio_processor;
};
```

20.27 AgsFxSynthAudioSignal

AgsFxSynthAudioSignal — fx synth audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSynthAudioSignal *	ags_fx_synth_audio_signal_new ()
#define	AGS_FX_SYNTH_AUDIO_SIGNAL()
#define	AGS_FX_SYNTH_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_SYNTH_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_SYNTH_AUDIO_SIGNAL()
#define	AGS_IS_FX_SYNTH_AUDIO_SIGNAL_CLASS()
GType	ags_fx_synth_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL
struct	AgsFxSynthAudioSignal
struct	AgsFxSynthAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioSignal
                &#x2570;&#x2500;&#x2500; AgsFxSynthAudioSignal
```

Implemented Interfaces

AgsFxSynthAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_synth_audio_signal.h>
```

Description

The AgsFxSynthAudioSignal class provides ports to the effect processor.

Functions

ags_fx_synth_audio_signal_new ()

```
AgsFxSynthAudioSignal~*
ags_fx_synth_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxSynthAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxSynthAudioSignal

Since: 3.14.0

AGS_FX_SYNTH_AUDIO_SIGNAL()

```
#define AGS_FX_SYNTH_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL, AgsFxSynthAudioSignal))
```

AGS_FX_SYNTH_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_SYNTH_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL, AgsFxSynthAudioSignalClass))
```

AGS_FX_SYNTH_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_SYNTH_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL, AgsFxSynthAudioSignalClass))
```

AGS_IS_FX_SYNTH_AUDIO_SIGNAL()

```
#define AGS_IS_FX_SYNTH_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL))
```

AGS_IS_FX_SYNTH_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_SYNTH_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL))
```

ags_fx_synth_audio_signal_get_type ()

```
GType
ags_fx_synth_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL                (ags_fx_synth_audio_signal_get_type() ↩
    )
```

struct AgsFxSynthAudioSignal

```
struct AgsFxSynthAudioSignal;
```

struct AgsFxSynthAudioSignalClass

```
struct AgsFxSynthAudioSignalClass {
  AgsFxNotationAudioSignalClass fx_notation_audio_signal;
};
```

20.28 AgsFxSynthChannel

AgsFxSynthChannel — fx synth channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSynthChannel *	ags_fx_synth_channel_new ()
#define	AGS_FX_SYNTH_CHANNEL()
#define	AGS_FX_SYNTH_CHANNEL_CLASS()
#define	AGS_FX_SYNTH_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_SYNTH_CHANNEL()
#define	AGS_IS_FX_SYNTH_CHANNEL_CLASS()
GType	ags_fx_synth_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_SYNTH_CHANNEL
struct	AgsFxSynthChannel
struct	AgsFxSynthChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannel
                &#x2570;&#x2500;&#x2500; AgsFxSynthChannel
```

Implemented Interfaces

AgsFxSynthChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_synth_channel.h>
```

Description

The AgsFxSynthChannel class provides ports to the effect processor.

Functions

ags_fx_synth_channel_new ()

```
AgsFxSynthChannel~*
ags_fx_synth_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxSynthChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxSynthChannel

Since: 3.14.0

AGS_FX_SYNTH_CHANNEL()

```
#define AGS_FX_SYNTH_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SYNTH_CHANNEL, AgsFxSynthChannel))
```

AGS_FX_SYNTH_CHANNEL_CLASS()

```
#define AGS_FX_SYNTH_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SYNTH_CHANNEL, AgsFxSynthChannelClass))
```

AGS_FX_SYNTH_CHANNEL_GET_CLASS()

```
#define AGS_FX_SYNTH_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SYNTH_CHANNEL, AgsFxSynthChannelClass))
```

AGS_IS_FX_SYNTH_CHANNEL()

```
#define AGS_IS_FX_SYNTH_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SYNTH_CHANNEL))
```

AGS_IS_FX_SYNTH_CHANNEL_CLASS()

```
#define AGS_IS_FX_SYNTH_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SYNTH_CHANNEL))
```

ags_fx_synth_channel_get_type ()

```
GType
ags_fx_synth_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_SYNTH_CHANNEL

```
#define AGS_TYPE_FX_SYNTH_CHANNEL                 (ags_fx_synth_channel_get_type())
```

struct AgsFxSynthChannel

```
struct AgsFxSynthChannel;
```

struct AgsFxSynthChannelClass

```
struct AgsFxSynthChannelClass {
  AgsFxNotationChannelClass fx_notation_channel;
};
```

20.29 AgsFxSynthChannelProcessor

AgsFxSynthChannelProcessor — fx synth channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSynthChannelProcessor *	ags_fx_synth_channel_processor_new ()
#define	AGS_FX_SYNTH_CHANNEL_PROCESSOR()
#define	AGS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_SYNTH_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_synth_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR
struct	AgsFxSynthChannelProcessor
struct	AgsFxSynthChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannelProcessor
                &#x2570;&#x2500;&#x2500; AgsFxSynthChannelProcessor
```

Implemented Interfaces

AgsFxSynthChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_synth_channel_processor.h>
```

Description

The AgsFxSynthChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_synth_channel_processor_new ()

```
AgsFxSynthChannelProcessor~*
ags_fx_synth_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxSynthChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxSynthChannelProcessor

Since: 3.14.0

AGS_FX_SYNTH_CHANNEL_PROCESSOR()

```
#define AGS_FX_SYNTH_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR, AgsFxSynthChannelProcessor))
```

AGS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class) ↩
    , AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR, AgsFxSynthChannelProcessorClass))
```

AGS_FX_SYNTH_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_SYNTH_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj ↩
    ), AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR, AgsFxSynthChannelProcessorClass))
```

AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR))
```

AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR))
```

ags_fx_synth_channel_processor_get_type ()

```
GType
ags_fx_synth_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR                ( ↩
    ags_fx_synth_channel_processor_get_type())
```

struct AgsFxSynthChannelProcessor

```
struct AgsFxSynthChannelProcessor;
```

struct AgsFxSynthChannelProcessorClass

```
struct AgsFxSynthChannelProcessorClass {
  AgsFxNotationChannelProcessorClass fx_notation_channel_processor;
};
```

20.30 AgsFxSynthRecycling

AgsFxSynthRecycling — fx synth recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSynthRecycling *	ags_fx_synth_recycling_new ()
#define	AGS_FX_SYNTH_RECYCLING()
#define	AGS_FX_SYNTH_RECYCLING_CLASS()
#define	AGS_FX_SYNTH_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_SYNTH_RECYCLING()
#define	AGS_IS_FX_SYNTH_RECYCLING_CLASS()
GType	ags_fx_synth_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_SYNTH_RECYCLING
struct	AgsFxSynthRecycling
struct	AgsFxSynthRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxNotationRecycling
                &#x2570;&#x2500;&#x2500; AgsFxSynthRecycling
```

Implemented Interfaces

AgsFxSynthRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_synth_recycling.h>
```

Description

The AgsFxSynthRecycling class provides ports to the effect processor.

Functions

ags_fx_synth_recycling_new ()

```
AgsFxSynthRecycling~*
ags_fx_synth_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxSynthRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxSynthRecycling

Since: 3.14.0

AGS_FX_SYNTH_RECYCLING()

```
#define AGS_FX_SYNTH_RECYCLING(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SYNTH_RECYCLING, AgsFxSynthRecycling))
```

AGS_FX_SYNTH_RECYCLING_CLASS()

```
#define AGS_FX_SYNTH_RECYCLING_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SYNTH_RECYCLING, AgsFxSynthRecyclingClass))
```

AGS_FX_SYNTH_RECYCLING_GET_CLASS()

```
#define AGS_FX_SYNTH_RECYCLING_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SYNTH_RECYCLING, AgsFxSynthRecyclingClass))
```

AGS_IS_FX_SYNTH_RECYCLING()

```
#define AGS_IS_FX_SYNTH_RECYCLING(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SYNTH_RECYCLING))
```

AGS_IS_FX_SYNTH_RECYCLING_CLASS()

```
#define AGS_IS_FX_SYNTH_RECYCLING_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SYNTH_RECYCLING))
```

ags_fx_synth_recycling_get_type ()

```
GType
ags_fx_synth_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_SYNTH_RECYCLING

```
#define AGS_TYPE_FX_SYNTH_RECYCLING                 (ags_fx_synth_recycling_get_type())
```

struct AgsFxSynthRecycling

```
struct AgsFxSynthRecycling;
```

struct AgsFxSynthRecyclingClass

```
struct AgsFxSynthRecyclingClass {
  AgsFxNotationRecyclingClass fx_notation_recycling;
};
```

20.31 AgsFxFMSynthAudio

AgsFxFMSynthAudio — fx FM synth audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxFMSynthAudioScopeData *	ags_fx_fm_synth_audio_scope_data_alloc ()
void	ags_fx_fm_synth_audio_scope_data_free ()
AgsFxFMSynthAudioChannelData *	ags_fx_fm_synth_audio_channel_data_alloc ()
void	ags_fx_fm_synth_audio_channel_data_free ()
AgsFxFMSynthAudioInputData *	ags_fx_fm_synth_audio_input_data_alloc ()
void	ags_fx_fm_synth_audio_input_data_free ()
AgsFxFMSynthAudio *	ags_fx_fm_synth_audio_new ()
#define	AGS_FX_FM_SYNTH_AUDIO()
#define	AGS_FX_FM_SYNTH_AUDIO_CLASS()
#define	AGS_FX_FM_SYNTH_AUDIO_GET_CLASS()
#define	AGS_IS_FX_FM_SYNTH_AUDIO()
#define	AGS_IS_FX_FM_SYNTH_AUDIO_CLASS()
GType	ags_fx_fm_synth_audio_get_type ()

Properties

AgsPort *	chorus-delay	Read / Write
AgsPort *	chorus-depth	Read / Write
AgsPort *	chorus-enabled	Read / Write
AgsPort *	chorus-input-volume	Read / Write
AgsPort *	chorus-lfo-frequency	Read / Write
AgsPort *	chorus-lfo-oscillator	Read / Write
AgsPort *	chorus-mix	Read / Write
AgsPort *	chorus-output-volume	Read / Write
AgsPort *	chorus-pitch-type	Read / Write
AgsPort *	high-pass-enabled	Read / Write
AgsPort *	high-pass-filter-gain	Read / Write
AgsPort *	high-pass-q-lin	Read / Write
AgsPort *	low-pass-enabled	Read / Write

AgsPort *	low-pass-filter-gain	Read / Write
AgsPort *	low-pass-q-lin	Read / Write
AgsPort *	noise-gain	Read / Write
AgsPort *	pitch-tuning	Read / Write
AgsPort *	pitch-type	Read / Write
AgsPort *	sequencer-enabled	Read / Write
AgsPort *	sequencer-sign	Read / Write
AgsPort *	synth-0-key	Read / Write
AgsPort *	synth-0-lfo-depth	Read / Write
AgsPort *	synth-0-lfo-frequency	Read / Write
AgsPort *	synth-0-lfo-oscillator	Read / Write
AgsPort *	synth-0-lfo-tuning	Read / Write
AgsPort *	synth-0-octave	Read / Write
AgsPort *	synth-0-oscillator	Read / Write
AgsPort *	synth-0-phase	Read / Write
AgsPort *	synth-0-volume	Read / Write
AgsPort *	synth-1-key	Read / Write
AgsPort *	synth-1-lfo-depth	Read / Write
AgsPort *	synth-1-lfo-frequency	Read / Write
AgsPort *	synth-1-lfo-oscillator	Read / Write
AgsPort *	synth-1-lfo-tuning	Read / Write
AgsPort *	synth-1-octave	Read / Write
AgsPort *	synth-1-oscillator	Read / Write
AgsPort *	synth-1-phase	Read / Write
AgsPort *	synth-1-volume	Read / Write
AgsPort *	synth-2-key	Read / Write
AgsPort *	synth-2-lfo-depth	Read / Write
AgsPort *	synth-2-lfo-frequency	Read / Write
AgsPort *	synth-2-lfo-oscillator	Read / Write
AgsPort *	synth-2-lfo-tuning	Read / Write
AgsPort *	synth-2-octave	Read / Write
AgsPort *	synth-2-oscillator	Read / Write
AgsPort *	synth-2-phase	Read / Write
AgsPort *	synth-2-volume	Read / Write

Types and Values

#define	AGS_FX_FM_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE
struct	AgsFxFMSynthAudioScopeData
struct	AgsFxFMSynthAudioChannelData
struct	AgsFxFMSynthAudioInputData
#define	AGS_TYPE_FX_FM_SYNTH_AUDIO
struct	AgsFxFMSynthAudio
struct	AgsFxFMSynthAudioClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudio
        &#x2570;&#x2500;&#x2500; AgsFxNotationAudio
            &#x2570;&#x2500;&#x2500; AgsFxFMSynthAudio
```

Implemented Interfaces

AgsFxFMSynthAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_fm_synth_audio.h>
```

Description

The AgsFxFMSynthAudio class provides ports to the effect processor.

Functions

ags_fx_fm_synth_audio_scope_data_alloc ()

```
AgsFxFMSynthAudioScopeData~*
ags_fx_fm_synth_audio_scope_data_alloc
                                ();
```

Allocate AgsFxFMSynthAudioScopeData

Returns

the new AgsFxFMSynthAudioScopeData

Since: 3.14.0

ags_fx_fm_synth_audio_scope_data_free ()

```
void
ags_fx_fm_synth_audio_scope_data_free (AgsFxFMSynthAudioScopeData *scope_data);
```

Free *scope_data* .

Parameters

scope_data	the AgsFxFMSynthAudioScopeData

Since: 3.14.0

ags_fx_fm_synth_audio_channel_data_alloc ()

```
AgsFxFMSynthAudioChannelData~*
ags_fx_fm_synth_audio_channel_data_alloc
                                ();
```

Allocate AgsFxFMSynthAudioChannelData

Returns

the new AgsFxFMSynthAudioChannelData

Since: 3.14.0

ags_fx_fm_synth_audio_channel_data_free ()

```
void
ags_fx_fm_synth_audio_channel_data_free
                                (AgsFxFMSynthAudioChannelData *channel_data);
```

Free *channel_data* .

Parameters

channel_data	the AgsFxFMSynthAudioChannelData

Since: 3.14.0

ags_fx_fm_synth_audio_input_data_alloc ()

```
AgsFxFMSynthAudioInputData~*
ags_fx_fm_synth_audio_input_data_alloc
                                ();
```

Allocate AgsFxFMSynthAudioInputData

Returns

the new AgsFxFMSynthAudioInputData

Since: 3.14.0

ags_fx_fm_synth_audio_input_data_free ()

```
void
ags_fx_fm_synth_audio_input_data_free (AgsFxFMSynthAudioInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxFMSynthAudioInputData

Since: 3.14.0

ags_fx_fm_synth_audio_new ()

```
AgsFxFMSynthAudio~*
ags_fx_fm_synth_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxFMSynthAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxFMSynthAudio

Since: 3.14.0

AGS_FX_FM_SYNTH_AUDIO()

```
#define AGS_FX_FM_SYNTH_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_AUDIO, AgsFxFMSynthAudio))
```

AGS_FX_FM_SYNTH_AUDIO_CLASS()

```
#define AGS_FX_FM_SYNTH_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_AUDIO, AgsFxFMSynthAudioClass))
```

AGS_FX_FM_SYNTH_AUDIO_GET_CLASS()

```
#define AGS_FX_FM_SYNTH_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_AUDIO, AgsFxFMSynthAudioClass))
```

AGS_IS_FX_FM_SYNTH_AUDIO()

```
#define AGS_IS_FX_FM_SYNTH_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_AUDIO))
```

AGS_IS_FX_FM_SYNTH_AUDIO_CLASS()

```
#define AGS_IS_FX_FM_SYNTH_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_AUDIO))
```

ags_fx_fm_synth_audio_get_type ()

```
GType
ags_fx_fm_synth_audio_get_type ();
```

Types and Values

AGS_FX_FM_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE

```
#define AGS_FX_FM_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE (65536)
```

struct AgsFxFMSynthAudioScopeData

```
struct AgsFxFMSynthAudioScopeData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint audio_channels;

  AgsFxFMSynthAudioChannelData **channel_data;
};
```

struct AgsFxFMSynthAudioChannelData

```
struct AgsFxFMSynthAudioChannelData {
  GRecMutex strct_mutex;

  gpointer parent;

  AgsFMSynthUtil *synth_0;
  AgsFMSynthUtil *synth_1;
  AgsFMSynthUtil *synth_2;

  AgsNoiseUtil *noise_util;

  GType pitch_type;
  gpointer pitch_util;

  AgsChorusUtil *chorus_util;

  AgsFluidIIRFilterUtil *low_pass_filter;
  AgsFluidIIRFilterUtil *high_pass_filter;

  AgsFxFMSynthAudioInputData* input_data[AGS_SEQUENCER_MAX_MIDI_KEYS];
};
```

struct AgsFxFMSynthAudioInputData

```
struct AgsFxFMSynthAudioInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint key_on;
};
```

AGS_TYPE_FX_FM_SYNTH_AUDIO

```
#define AGS_TYPE_FX_FM_SYNTH_AUDIO                (ags_fx_fm_synth_audio_get_type())
```

struct AgsFxFMSynthAudio

```
struct AgsFxFMSynthAudio;
```

struct AgsFxFMSynthAudioClass

```
struct AgsFxFMSynthAudioClass {
  AgsFxNotationAudioClass fx_notation_audio;
};
```

Property Details

The "chorus-delay" property

```
  "chorus-delay"              AgsPort~*
```

The chorus delay.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-depth" property

```
  "chorus-depth"              AgsPort~*
```

The chorus depth.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-enabled" property

```
  "chorus-enabled"            AgsPort~*
```

The chorus enabled.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-input-volume" property

```
  "chorus-input-volume"       AgsPort~*
```

The chorus input volume.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-lfo-frequency" property

```
"chorus-lfo-frequency"      AgsPort~*
```

The chorus lfo frequency.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-lfo-oscillator" property

```
"chorus-lfo-oscillator"     AgsPort~*
```

The chorus lfo oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-mix" property

```
"chorus-mix"                AgsPort~*
```

The chorus mix.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-output-volume" property

```
"chorus-output-volume"      AgsPort~*
```

The chorus output volume.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "chorus-pitch-type" property

```
"chorus-pitch-type"         AgsPort~*
```

The chorus pitch type.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 4.0.0

The "high-pass-enabled" property

```
"high-pass-enabled"        AgsPort~*
```

The high pass enabled.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "high-pass-filter-gain" property

```
"high-pass-filter-gain"    AgsPort~*
```

The high pass filter gain.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "high-pass-q-lin" property

```
"high-pass-q-lin"          AgsPort~*
```

The high pass q-lin.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "low-pass-enabled" property

```
"low-pass-enabled"         AgsPort~*
```

The low pass enabled.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "low-pass-filter-gain" property

```
"low-pass-filter-gain"     AgsPort~*
```

The low pass filter gain.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "low-pass-q-lin" property

```
"low-pass-q-lin"            AgsPort~*
```

The low pass q-lin.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "noise-gain" property

```
"noise-gain"                AgsPort~*
```

The noise gain.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "pitch-tuning" property

```
"pitch-tuning"              AgsPort~*
```

The pitch tuning.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "pitch-type" property

```
"pitch-type"                AgsPort~*
```

The pitch type.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 4.0.0

The "sequencer-enabled" property

```
"sequencer-enabled"         AgsPort~*
```

The sequencer enabled.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "sequencer-sign" property

```
"sequencer-sign"            AgsPort~*
```

The sequencer sign.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-key" property

```
"synth-0-key"               AgsPort~*
```

The synth-0 key.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-lfo-depth" property

```
"synth-0-lfo-depth"         AgsPort~*
```

The synth-0 LFO depth.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-lfo-frequency" property

```
"synth-0-lfo-frequency"     AgsPort~*
```

The synth-0 LFO frequency.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-lfo-oscillator" property

```
"synth-0-lfo-oscillator"    AgsPort~*
```

The synth-0 lfo-oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-lfo-tuning" property

```
"synth-0-lfo-tuning"        AgsPort~*
```

The synth-0 LFO tuning.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-octave" property

```
"synth-0-octave"            AgsPort~*
```

The synth-0 octave.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-oscillator" property

```
"synth-0-oscillator"        AgsPort~*
```

The synth-0 oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-phase" property

```
"synth-0-phase"             AgsPort~*
```

The synth-0 phase.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-0-volume" property

```
"synth-0-volume"            AgsPort~*
```

The synth-0 volume.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-key" property

```
"synth-1-key"              AgsPort~*
```

The synth-1 key.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-lfo-depth" property

```
"synth-1-lfo-depth"        AgsPort~*
```

The synth-1 LFO depth.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-lfo-frequency" property

```
"synth-1-lfo-frequency"    AgsPort~*
```

The synth-1 LFO frequency.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-lfo-oscillator" property

```
"synth-1-lfo-oscillator"   AgsPort~*
```

The synth-1 lfo-oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-lfo-tuning" property

```
"synth-1-lfo-tuning"       AgsPort~*
```

The synth-1 LFO tuning.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-octave" property

```
"synth-1-octave"            AgsPort~*
```

The synth-1 octave.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-oscillator" property

```
"synth-1-oscillator"        AgsPort~*
```

The synth-1 oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-phase" property

```
"synth-1-phase"             AgsPort~*
```

The synth-1 phase.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-1-volume" property

```
"synth-1-volume"            AgsPort~*
```

The synth-1 volume.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-key" property

```
"synth-2-key"               AgsPort~*
```

The synth-2 key.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-lfo-depth" property

```
"synth-2-lfo-depth"        AgsPort~*
```

The synth-2 LFO depth.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-lfo-frequency" property

```
"synth-2-lfo-frequency"    AgsPort~*
```

The synth-2 LFO frequency.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-lfo-oscillator" property

```
"synth-2-lfo-oscillator"   AgsPort~*
```

The synth-2 lfo-oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-lfo-tuning" property

```
"synth-2-lfo-tuning"       AgsPort~*
```

The synth-2 LFO tuning.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-octave" property

```
"synth-2-octave"           AgsPort~*
```

The synth-2 octave.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-oscillator" property

```
"synth-2-oscillator"        AgsPort~*
```

The synth-2 oscillator.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-phase" property

```
"synth-2-phase"             AgsPort~*
```

The synth-2 phase.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

The "synth-2-volume" property

```
"synth-2-volume"            AgsPort~*
```

The synth-2 volume.

Owner: AgsFxFMSynthAudio

Flags: Read / Write

Since: 3.15.0

20.32 AgsFxFMSynthAudioProcessor

AgsFxFMSynthAudioProcessor — fx FM synth audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxFMSynthAudioProcessor *	ags_fx_fm_synth_audio_processor_new ()
#define	AGS_FX_FM_SYNTH_AUDIO_PROCESSOR()
#define	AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR()
#define	AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_fm_synth_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR
struct	AgsFxFMSynthAudioProcessor
struct	AgsFxFMSynthAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioProcessor
                &#x2570;&#x2500;&#x2500; AgsFxFMSynthAudioProcessor
```

Implemented Interfaces

AgsFxFMSynthAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_fm_synth_audio_processor.h>
```

Description

The AgsFxFMSynthAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_fm_synth_audio_processor_new ()

```
AgsFxFMSynthAudioProcessor~*
ags_fx_fm_synth_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxFMSynthAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxFMSynthAudioProcessor

Since: 3.14.0

AGS_FX_FM_SYNTH_AUDIO_PROCESSOR()

```
#define AGS_FX_FM_SYNTH_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR, AgsFxFMSynthAudioProcessor))
```

AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR, AgsFxFMSynthAudioProcessorClass))
```

AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR, AgsFxFMSynthAudioProcessorClass))
```

AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR))
```

AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR))
```

ags_fx_fm_synth_audio_processor_get_type ()

```
GType
ags_fx_fm_synth_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR                ( ↩
    ags_fx_fm_synth_audio_processor_get_type())
```

struct AgsFxFMSynthAudioProcessor

```
struct AgsFxFMSynthAudioProcessor;
```

struct AgsFxFMSynthAudioProcessorClass

```
struct AgsFxFMSynthAudioProcessorClass {
  AgsFxNotationAudioProcessorClass fx_notation_audio_processor;
};
```

20.33 AgsFxFMSynthAudioSignal

AgsFxFMSynthAudioSignal — fx FM synth audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxFMSynthAudioSignal *	ags_fx_fm_synth_audio_signal_new ()
#define	AGS_FX_FM_SYNTH_AUDIO_SIGNAL()
#define	AGS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_FM_SYNTH_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL()
#define	AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS()
GType	ags_fx_fm_synth_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL
struct	AgsFxFMSynthAudioSignal
struct	AgsFxFMSynthAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioSignal
                &#x2570;&#x2500;&#x2500; AgsFxFMSynthAudioSignal
```

Implemented Interfaces

AgsFxFMSynthAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_fm_synth_audio_signal.h>
```

Description

The AgsFxFMSynthAudioSignal class provides ports to the effect processor.

Functions

ags_fx_fm_synth_audio_signal_new ()

```
AgsFxFMSynthAudioSignal~*
ags_fx_fm_synth_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxFMSynthAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxFMSynthAudioSignal

Since: 3.14.0

AGS_FX_FM_SYNTH_AUDIO_SIGNAL()

```
#define AGS_FX_FM_SYNTH_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL, AgsFxFMSynthAudioSignal))
```

AGS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL, AgsFxFMSynthAudioSignalClass))
```

AGS_FX_FM_SYNTH_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_FM_SYNTH_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL, AgsFxFMSynthAudioSignalClass))
```

AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL()

```
#define AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL))
```

AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL))
```

ags_fx_fm_synth_audio_signal_get_type ()

```
GType
ags_fx_fm_synth_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL                ( ↩
    ags_fx_fm_synth_audio_signal_get_type())
```

struct AgsFxFMSynthAudioSignal

```
struct AgsFxFMSynthAudioSignal;
```

struct AgsFxFMSynthAudioSignalClass

```
struct AgsFxFMSynthAudioSignalClass {
  AgsFxNotationAudioSignalClass fx_notation_audio_signal;
};
```

20.34 AgsFxFMSynthChannel

AgsFxFMSynthChannel — fx FM synth channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxFMSynthChannel *	ags_fx_fm_synth_channel_new ()
#define	AGS_FX_FM_SYNTH_CHANNEL()
#define	AGS_FX_FM_SYNTH_CHANNEL_CLASS()
#define	AGS_FX_FM_SYNTH_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_FM_SYNTH_CHANNEL()
#define	AGS_IS_FX_FM_SYNTH_CHANNEL_CLASS()
GType	ags_fx_fm_synth_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_FM_SYNTH_CHANNEL
struct	AgsFxFMSynthChannel
struct	AgsFxFMSynthChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannel
                &#x2570;&#x2500;&#x2500; AgsFxFMSynthChannel
```

Implemented Interfaces

AgsFxFMSynthChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_fm_synth_channel.h>
```

Description

The AgsFxFMSynthChannel class provides ports to the effect processor.

Functions

ags_fx_fm_synth_channel_new ()

```
AgsFxFMSynthChannel~*
ags_fx_fm_synth_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxFMSynthChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxFMSynthChannel

Since: 3.14.0

AGS_FX_FM_SYNTH_CHANNEL()

```
#define AGS_FX_FM_SYNTH_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_CHANNEL, AgsFxFMSynthChannel))
```

AGS_FX_FM_SYNTH_CHANNEL_CLASS()

```
#define AGS_FX_FM_SYNTH_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_CHANNEL, AgsFxFMSynthChannelClass))
```

AGS_FX_FM_SYNTH_CHANNEL_GET_CLASS()

```
#define AGS_FX_FM_SYNTH_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_CHANNEL, AgsFxFMSynthChannelClass))
```

AGS_IS_FX_FM_SYNTH_CHANNEL()

```
#define AGS_IS_FX_FM_SYNTH_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_CHANNEL))
```

AGS_IS_FX_FM_SYNTH_CHANNEL_CLASS()

```
#define AGS_IS_FX_FM_SYNTH_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_CHANNEL))
```

ags_fx_fm_synth_channel_get_type ()

```
GType
ags_fx_fm_synth_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_FM_SYNTH_CHANNEL

```
#define AGS_TYPE_FX_FM_SYNTH_CHANNEL                 (ags_fx_fm_synth_channel_get_type())
```

struct AgsFxFMSynthChannel

```
struct AgsFxFMSynthChannel;
```

struct AgsFxFMSynthChannelClass

```
struct AgsFxFMSynthChannelClass {
  AgsFxNotationChannelClass fx_notation_channel;
};
```

20.35 AgsFxFMSynthChannelProcessor

AgsFxFMSynthChannelProcessor — fx FM synth channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxFMSynthChannelProcessor *	ags_fx_fm_synth_channel_processor_new ()
#define	AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR()
#define	AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_fm_synth_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR
struct	AgsFxFMSynthChannelProcessor
struct	AgsFxFMSynthChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannelProcessor
                &#x2570;&#x2500;&#x2500; AgsFxFMSynthChannelProcessor
```

Implemented Interfaces

AgsFxFMSynthChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_fm_synth_channel_processor.h>
```

Description

The AgsFxFMSynthChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_fm_synth_channel_processor_new ()

```
AgsFxFMSynthChannelProcessor~*
ags_fx_fm_synth_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxFMSynthChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxFMSynthChannelProcessor

Since: 3.14.0

AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR()

```
#define AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR, AgsFxFMSynthChannelProcessor))
```

AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR, AgsFxFMSynthChannelProcessorClass))
```

AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR, AgsFxFMSynthChannelProcessorClass))
```

AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR))
```

AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR))
```

ags_fx_fm_synth_channel_processor_get_type ()

```
GType
ags_fx_fm_synth_channel_processor_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR                ( ↩
    ags_fx_fm_synth_channel_processor_get_type())
```

struct AgsFxFMSynthChannelProcessor

```
struct AgsFxFMSynthChannelProcessor;
```

struct AgsFxFMSynthChannelProcessorClass

```
struct AgsFxFMSynthChannelProcessorClass {
  AgsFxNotationChannelProcessorClass fx_notation_channel_processor;
};
```

20.36 AgsFxFMSynthRecycling

AgsFxFMSynthRecycling — fx FM synth recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxFMSynthRecycling *	ags_fx_fm_synth_recycling_new ()
#define	AGS_FX_FM_SYNTH_RECYCLING()
#define	AGS_FX_FM_SYNTH_RECYCLING_CLASS()
#define	AGS_FX_FM_SYNTH_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_FM_SYNTH_RECYCLING()
#define	AGS_IS_FX_FM_SYNTH_RECYCLING_CLASS()
GType	ags_fx_fm_synth_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_FM_SYNTH_RECYCLING
struct	AgsFxFMSynthRecycling
struct	AgsFxFMSynthRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxNotationRecycling
                &#x2570;&#x2500;&#x2500; AgsFxFMSynthRecycling
```

Implemented Interfaces

AgsFxFMSynthRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_fm_synth_recycling.h>
```

Description

The AgsFxFMSynthRecycling class provides ports to the effect processor.

Functions

ags_fx_fm_synth_recycling_new ()

```
AgsFxFMSynthRecycling~*
ags_fx_fm_synth_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxFMSynthRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxFMSynthRecycling

Since: 3.14.0

AGS_FX_FM_SYNTH_RECYCLING()

```
#define AGS_FX_FM_SYNTH_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_RECYCLING, AgsFxFMSynthRecycling))
```

AGS_FX_FM_SYNTH_RECYCLING_CLASS()

```
#define AGS_FX_FM_SYNTH_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_RECYCLING, AgsFxFMSynthRecyclingClass))
```

AGS_FX_FM_SYNTH_RECYCLING_GET_CLASS()

```
#define AGS_FX_FM_SYNTH_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_RECYCLING, AgsFxFMSynthRecyclingClass))
```

AGS_IS_FX_FM_SYNTH_RECYCLING()

```
#define AGS_IS_FX_FM_SYNTH_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_FM_SYNTH_RECYCLING))
```

AGS_IS_FX_FM_SYNTH_RECYCLING_CLASS()

```
#define AGS_IS_FX_FM_SYNTH_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_FM_SYNTH_RECYCLING))
```

ags_fx_fm_synth_recycling_get_type ()

```
GType
ags_fx_fm_synth_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_FM_SYNTH_RECYCLING

```
#define AGS_TYPE_FX_FM_SYNTH_RECYCLING                (ags_fx_fm_synth_recycling_get_type() ↩
    )
```

struct AgsFxFMSynthRecycling

```
struct AgsFxFMSynthRecycling;
```

struct AgsFxFMSynthRecyclingClass

```
struct AgsFxFMSynthRecyclingClass {
  AgsFxNotationRecyclingClass fx_notation_recycling;
};
```

20.37 AgsFxSF2SynthAudio

AgsFxSF2SynthAudio — fx Soundfont2 synth audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSF2SynthAudioScopeData *	ags_fx_sf2_synth_audio_scope_data_alloc ()
void	ags_fx_sf2_synth_audio_scope_data_free ()
AgsFxSF2SynthAudioChannelData *	ags_fx_sf2_synth_audio_channel_data_alloc ()
void	ags_fx_sf2_synth_audio_channel_data_free ()
AgsFxSF2SynthAudioInputData *	ags_fx_sf2_synth_audio_input_data_alloc ()
void	ags_fx_sf2_synth_audio_input_data_free ()
AgsFxSF2SynthAudio *	ags_fx_sf2_synth_audio_new ()
#define	AGS_FX_SF2_SYNTH_AUDIO()
#define	AGS_FX_SF2_SYNTH_AUDIO_CLASS()
#define	AGS_FX_SF2_SYNTH_AUDIO_GET_CLASS()
#define	AGS_IS_FX_SF2_SYNTH_AUDIO()
#define	AGS_IS_FX_SF2_SYNTH_AUDIO_CLASS()
GType	ags_fx_sf2_synth_audio_get_type ()

Properties

AgsPort *	chorus-delay	Read / Write
AgsPort *	chorus-depth	Read / Write
AgsPort *	chorus-enabled	Read / Write
AgsPort *	chorus-input-volume	Read / Write
AgsPort *	chorus-lfo-frequency	Read / Write
AgsPort *	chorus-lfo-oscillator	Read / Write
AgsPort *	chorus-mix	Read / Write
AgsPort *	chorus-output-volume	Read / Write
AgsPort *	chorus-pitch-type	Read / Write
AgsPort *	synth-key	Read / Write
AgsPort *	synth-octave	Read / Write
AgsPort *	synth-pitch-type	Read / Write
AgsPort *	synth-volume	Read / Write

Types and Values

#define	AGS_FX_SF2_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE
struct	AgsFxSF2SynthAudioScopeData
struct	AgsFxSF2SynthAudioChannelData
struct	AgsFxSF2SynthAudioInputData
#define	AGS_TYPE_FX_SF2_SYNTH_AUDIO
struct	AgsFxSF2SynthAudio
struct	AgsFxSF2SynthAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudio
                &#x2570;&#x2500;&#x2500; AgsFxSF2SynthAudio
```

Implemented Interfaces

AgsFxSF2SynthAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_sf2_synth_audio.h>
```

Description

The AgsFxSF2SynthAudio class provides ports to the effect processor.

Functions

ags_fx_sf2_synth_audio_scope_data_alloc ()

```
AgsFxSF2SynthAudioScopeData~*
ags_fx_sf2_synth_audio_scope_data_alloc
                                ();
```

Allocate AgsFxSF2SynthAudioScopeData

Returns

the new AgsFxSF2SynthAudioScopeData

Since: 3.16.0

ags_fx_sf2_synth_audio_scope_data_free ()

```
void
ags_fx_sf2_synth_audio_scope_data_free
                                (AgsFxSF2SynthAudioScopeData *scope_data);
```

Free *scope_data* .

Parameters

scope_data	the AgsFxSF2SynthAudioScopeData

Since: 3.16.0

ags_fx_sf2_synth_audio_channel_data_alloc ()

```
AgsFxSF2SynthAudioChannelData~*
ags_fx_sf2_synth_audio_channel_data_alloc
                                ();
```

Allocate AgsFxSF2SynthAudioChannelData

Returns

the new AgsFxSF2SynthAudioChannelData

Since: 3.16.0

ags_fx_sf2_synth_audio_channel_data_free ()

```
void
ags_fx_sf2_synth_audio_channel_data_free
                                (AgsFxSF2SynthAudioChannelData *channel_data);
```

Free *channel_data* .

Parameters

channel_data	the AgsFxSF2SynthAudioChannelData

Since: 3.16.0

ags_fx_sf2_synth_audio_input_data_alloc ()

```
AgsFxSF2SynthAudioInputData~*
ags_fx_sf2_synth_audio_input_data_alloc
                                ();
```

Allocate AgsFxSF2SynthAudioInputData

Returns

the new AgsFxSF2SynthAudioInputData

Since: 3.16.0

ags_fx_sf2_synth_audio_input_data_free ()

```
void
ags_fx_sf2_synth_audio_input_data_free
                                (AgsFxSF2SynthAudioInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxSF2SynthAudioInputData

Since: 3.16.0

ags_fx_sf2_synth_audio_new ()

```
AgsFxSF2SynthAudio~*
ags_fx_sf2_synth_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxSF2SynthAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxSF2SynthAudio

Since: 3.16.0

AGS_FX_SF2_SYNTH_AUDIO()

```
#define AGS_FX_SF2_SYNTH_AUDIO(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_AUDIO, AgsFxSF2SynthAudio))
```

AGS_FX_SF2_SYNTH_AUDIO_CLASS()

```
#define AGS_FX_SF2_SYNTH_AUDIO_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SF2_SYNTH_AUDIO, AgsFxSF2SynthAudioClass))
```

AGS_FX_SF2_SYNTH_AUDIO_GET_CLASS()

```
#define AGS_FX_SF2_SYNTH_AUDIO_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_AUDIO, AgsFxSF2SynthAudioClass))
```

AGS_IS_FX_SF2_SYNTH_AUDIO()

```
#define AGS_IS_FX_SF2_SYNTH_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_AUDIO))
```

AGS_IS_FX_SF2_SYNTH_AUDIO_CLASS()

```
#define AGS_IS_FX_SF2_SYNTH_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SF2_SYNTH_AUDIO))
```

ags_fx_sf2_synth_audio_get_type ()

```
GType
ags_fx_sf2_synth_audio_get_type ();
```

Types and Values

AGS_FX_SF2_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE

```
#define AGS_FX_SF2_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE (65536)
```

struct AgsFxSF2SynthAudioScopeData

```
struct AgsFxSF2SynthAudioScopeData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint audio_channels;

  AgsFxSF2SynthAudioChannelData **channel_data;
};
```

struct AgsFxSF2SynthAudioChannelData

```
struct AgsFxSF2SynthAudioChannelData {
  GRecMutex strct_mutex;

  gpointer parent;

  AgsSF2SynthUtil *synth;

  AgsChorusUtil *chorus_util;

  AgsFxSF2SynthAudioInputData* input_data[AGS_SEQUENCER_MAX_MIDI_KEYS];
};
```

struct AgsFxSF2SynthAudioInputData

```
struct AgsFxSF2SynthAudioInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  guint key_on;
};
```

AGS_TYPE_FX_SF2_SYNTH_AUDIO

```
#define AGS_TYPE_FX_SF2_SYNTH_AUDIO                (ags_fx_sf2_synth_audio_get_type())
```

struct AgsFxSF2SynthAudio

```
struct AgsFxSF2SynthAudio;
```

struct AgsFxSF2SynthAudioClass

```
struct AgsFxSF2SynthAudioClass {
  AgsFxNotationAudioClass fx_notation_audio;
};
```

Property Details

The "chorus-delay" property

```
  "chorus-delay"             AgsPort~*
```

The chorus delay.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-depth" property

```
  "chorus-depth"             AgsPort~*
```

The chorus depth.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-enabled" property

```
"chorus-enabled"            AgsPort~*
```

The chorus enabled.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-input-volume" property

```
"chorus-input-volume"       AgsPort~*
```

The chorus input volume.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-lfo-frequency" property

```
"chorus-lfo-frequency"      AgsPort~*
```

The chorus lfo frequency.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-lfo-oscillator" property

```
"chorus-lfo-oscillator"     AgsPort~*
```

The chorus lfo oscillator.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-mix" property

```
"chorus-mix"                AgsPort~*
```

The chorus mix.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-output-volume" property

```
"chorus-output-volume"      AgsPort~*
```

The chorus output volume.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "chorus-pitch-type" property

```
"chorus-pitch-type"         AgsPort~*
```

The chorus's pitch type.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 4.0.0

The "synth-key" property

```
"synth-key"                 AgsPort~*
```

The synth key.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "synth-octave" property

```
"synth-octave"              AgsPort~*
```

The synth octave.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

The "synth-pitch-type" property

```
"synth-pitch-type"          AgsPort~*
```

The synth's pitch type.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 4.0.0

The "synth-volume" property

```
"synth-volume"              AgsPort~*
```

The synth volume.

Owner: AgsFxSF2SynthAudio

Flags: Read / Write

Since: 3.16.0

20.38 AgsFxSF2SynthAudioProcessor

AgsFxSF2SynthAudioProcessor — fx Soundfont2 synth audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSF2SynthAudioProcessor *	ags_fx_sf2_synth_audio_processor_new ()
#define	AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR()
#define	AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR()
#define	AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_sf2_synth_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR
struct	AgsFxSF2SynthAudioProcessor
struct	AgsFxSF2SynthAudioProcessorClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
        &#x2570;&#x2500;&#x2500; AgsFxNotationAudioProcessor
            &#x2570;&#x2500;&#x2500; AgsFxSF2SynthAudioProcessor
```

Implemented Interfaces

AgsFxSF2SynthAudioProcessor implements AgsConnectable, AgsCountable, AgsSeekable and AgsTactable.

Includes

```
#include <ags/audio/fx/ags_fx_sf2_synth_audio_processor.h>
```

Description

The AgsFxSF2SynthAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_sf2_synth_audio_processor_new ()

```
AgsFxSF2SynthAudioProcessor~*
ags_fx_sf2_synth_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxSF2SynthAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxSF2SynthAudioProcessor

Since: 3.16.0

AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR()

```
#define AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR, AgsFxSF2SynthAudioProcessor))
```

AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR, AgsFxSF2SynthAudioProcessorClass))
```

AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR, AgsFxSF2SynthAudioProcessorClass))
```

AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR))
```

AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR))
```

ags_fx_sf2_synth_audio_processor_get_type ()

```
GType
ags_fx_sf2_synth_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR                ( ↩
    ags_fx_sf2_synth_audio_processor_get_type())
```

struct AgsFxSF2SynthAudioProcessor

```
struct AgsFxSF2SynthAudioProcessor;
```

struct AgsFxSF2SynthAudioProcessorClass

```
struct AgsFxSF2SynthAudioProcessorClass {
  AgsFxNotationAudioProcessorClass fx_notation_audio_processor;
};
```

20.39 AgsFxSF2SynthAudioSignal

AgsFxSF2SynthAudioSignal — fx Soundfont2 synth audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSF2SynthAudioSignal *	ags_fx_sf2_synth_audio_signal_new ()
#define	AGS_FX_SF2_SYNTH_AUDIO_SIGNAL()
#define	AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL()
#define	AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS()
GType	ags_fx_sf2_synth_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL
struct	AgsFxSF2SynthAudioSignal
struct	AgsFxSF2SynthAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxNotationAudioSignal
                &#x2570;&#x2500;&#x2500; AgsFxSF2SynthAudioSignal
```

Implemented Interfaces

AgsFxSF2SynthAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_sf2_synth_audio_signal.h>
```

Description

The AgsFxSF2SynthAudioSignal class provides ports to the effect processor.

Functions

ags_fx_sf2_synth_audio_signal_new ()

```
AgsFxSF2SynthAudioSignal~*
ags_fx_sf2_synth_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxSF2SynthAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxSF2SynthAudioSignal

Since: 3.16.0

AGS_FX_SF2_SYNTH_AUDIO_SIGNAL()

```
#define AGS_FX_SF2_SYNTH_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL, AgsFxSF2SynthAudioSignal))
```

AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL, AgsFxSF2SynthAudioSignalClass))
```

AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL, AgsFxSF2SynthAudioSignalClass))
```

AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL()

```
#define AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL))
```

AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL))
```

ags_fx_sf2_synth_audio_signal_get_type ()

```
GType
ags_fx_sf2_synth_audio_signal_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL                ( ↩
    ags_fx_sf2_synth_audio_signal_get_type())
```

struct AgsFxSF2SynthAudioSignal

```
struct AgsFxSF2SynthAudioSignal;
```

struct AgsFxSF2SynthAudioSignalClass

```
struct AgsFxSF2SynthAudioSignalClass {
  AgsFxNotationAudioSignalClass fx_notation_audio_signal;
};
```

20.40 AgsFxSF2SynthChannel

AgsFxSF2SynthChannel — fx Soundfont2 synth channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSF2SynthChannel *	ags_fx_sf2_synth_channel_new ()
#define	AGS_FX_SF2_SYNTH_CHANNEL()
#define	AGS_FX_SF2_SYNTH_CHANNEL_CLASS()
#define	AGS_FX_SF2_SYNTH_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_SF2_SYNTH_CHANNEL()
#define	AGS_IS_FX_SF2_SYNTH_CHANNEL_CLASS()
GType	ags_fx_sf2_synth_channel_get_type ()

Types and Values

#define	AGS_TYPE_FX_SF2_SYNTH_CHANNEL
struct	AgsFxSF2SynthChannel
struct	AgsFxSF2SynthChannelClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannel
        &#x2570;&#x2500;&#x2500; AgsFxNotationChannel
            &#x2570;&#x2500;&#x2500; AgsFxSF2SynthChannel
```

Implemented Interfaces

AgsFxSF2SynthChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_sf2_synth_channel.h>
```

Description

The AgsFxSF2SynthChannel class provides ports to the effect processor.

Functions

ags_fx_sf2_synth_channel_new ()

```
AgsFxSF2SynthChannel~*
ags_fx_sf2_synth_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxSF2SynthChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxSF2SynthChannel

Since: 3.16.0

AGS_FX_SF2_SYNTH_CHANNEL()

```
#define AGS_FX_SF2_SYNTH_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_CHANNEL, AgsFxSF2SynthChannel))
```

AGS_FX_SF2_SYNTH_CHANNEL_CLASS()

```
#define AGS_FX_SF2_SYNTH_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SF2_SYNTH_CHANNEL, AgsFxSF2SynthChannelClass))
```

AGS_FX_SF2_SYNTH_CHANNEL_GET_CLASS()

```
#define AGS_FX_SF2_SYNTH_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_CHANNEL, AgsFxSF2SynthChannelClass))
```

AGS_IS_FX_SF2_SYNTH_CHANNEL()

```
#define AGS_IS_FX_SF2_SYNTH_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_CHANNEL))
```

AGS_IS_FX_SF2_SYNTH_CHANNEL_CLASS()

```
#define AGS_IS_FX_SF2_SYNTH_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SF2_SYNTH_CHANNEL))
```

ags_fx_sf2_synth_channel_get_type ()

```
GType
ags_fx_sf2_synth_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_SF2_SYNTH_CHANNEL

```
#define AGS_TYPE_FX_SF2_SYNTH_CHANNEL                (ags_fx_sf2_synth_channel_get_type())
```

struct AgsFxSF2SynthChannel

```
struct AgsFxSF2SynthChannel;
```

struct AgsFxSF2SynthChannelClass

```
struct AgsFxSF2SynthChannelClass {
  AgsFxNotationChannelClass fx_notation_channel;
};
```

20.41 AgsFxSF2SynthChannelProcessor

AgsFxSF2SynthChannelProcessor — fx Soundfont2 synth channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSF2SynthChannelProcessor *	ags_fx_sf2_synth_channel_processor_new ()
#define	AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR()
#define	AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_sf2_synth_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR
struct	AgsFxSF2SynthChannelProcessor
struct	AgsFxSF2SynthChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxNotationChannelProcessor
                &#x2570;&#x2500;&#x2500; AgsFxSF2SynthChannelProcessor
```

Implemented Interfaces

AgsFxSF2SynthChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_sf2_synth_channel_processor.h>
```

Description

The AgsFxSF2SynthChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_sf2_synth_channel_processor_new ()

```
AgsFxSF2SynthChannelProcessor~*
ags_fx_sf2_synth_channel_processor_new
                                (AgsChannel *channel);
```

Create a new instance of AgsFxSF2SynthChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxSF2SynthChannelProcessor

Since: 3.16.0

AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR()

```
#define AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST ↩
    ((obj), AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR, AgsFxSF2SynthChannelProcessor))
```

AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR, AgsFxSF2SynthChannelProcessorClass))
```

AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS  ↩
    ((obj), AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR, AgsFxSF2SynthChannelProcessorClass))
```

AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR))
```

AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR))
```

ags_fx_sf2_synth_channel_processor_get_type ()

```
GType
ags_fx_sf2_synth_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR                ( ↩
    ags_fx_sf2_synth_channel_processor_get_type())
```

struct AgsFxSF2SynthChannelProcessor

```
struct AgsFxSF2SynthChannelProcessor;
```

struct AgsFxSF2SynthChannelProcessorClass

```
struct AgsFxSF2SynthChannelProcessorClass {
  AgsFxNotationChannelProcessorClass fx_notation_channel_processor;
};
```

20.42 AgsFxSF2SynthRecycling

AgsFxSF2SynthRecycling — fx Soundfont2 synth recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxSF2SynthRecycling *	ags_fx_sf2_synth_recycling_new ()
#define	AGS_FX_SF2_SYNTH_RECYCLING()
#define	AGS_FX_SF2_SYNTH_RECYCLING_CLASS()
#define	AGS_FX_SF2_SYNTH_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_SF2_SYNTH_RECYCLING()
#define	AGS_IS_FX_SF2_SYNTH_RECYCLING_CLASS()
GType	ags_fx_sf2_synth_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_SF2_SYNTH_RECYCLING
struct	AgsFxSF2SynthRecycling
struct	AgsFxSF2SynthRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxNotationRecycling
                &#x2570;&#x2500;&#x2500; AgsFxSF2SynthRecycling
```

Implemented Interfaces

AgsFxSF2SynthRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_sf2_synth_recycling.h>
```

Description

The AgsFxSF2SynthRecycling class provides ports to the effect processor.

Functions

ags_fx_sf2_synth_recycling_new ()

```
AgsFxSF2SynthRecycling~*
ags_fx_sf2_synth_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxSF2SynthRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxSF2SynthRecycling

Since: 3.16.0

AGS_FX_SF2_SYNTH_RECYCLING()

```
#define AGS_FX_SF2_SYNTH_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_RECYCLING, AgsFxSF2SynthRecycling))
```

AGS_FX_SF2_SYNTH_RECYCLING_CLASS()

```
#define AGS_FX_SF2_SYNTH_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_SF2_SYNTH_RECYCLING, AgsFxSF2SynthRecyclingClass))
```

AGS_FX_SF2_SYNTH_RECYCLING_GET_CLASS()

```
#define AGS_FX_SF2_SYNTH_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_RECYCLING, AgsFxSF2SynthRecyclingClass))
```

AGS_IS_FX_SF2_SYNTH_RECYCLING()

```
#define AGS_IS_FX_SF2_SYNTH_RECYCLING(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_SF2_SYNTH_RECYCLING))
```

AGS_IS_FX_SF2_SYNTH_RECYCLING_CLASS()

```
#define AGS_IS_FX_SF2_SYNTH_RECYCLING_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_SF2_SYNTH_RECYCLING))
```

ags_fx_sf2_synth_recycling_get_type ()

```
GType
ags_fx_sf2_synth_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_SF2_SYNTH_RECYCLING

```
#define AGS_TYPE_FX_SF2_SYNTH_RECYCLING                 (ags_fx_sf2_synth_recycling_get_type ↩
    ())
```

struct AgsFxSF2SynthRecycling

```
struct AgsFxSF2SynthRecycling;
```

struct AgsFxSF2SynthRecyclingClass

```
struct AgsFxSF2SynthRecyclingClass {
  AgsFxNotationRecyclingClass fx_notation_recycling;
};
```

Chapter 21

Fx Engine - peak and analyse

Peak detection and analyse spectrum of audio data.

21.1 AgsFxPeakAudio

AgsFxPeakAudio — fx peak audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPeakAudio *	ags_fx_peak_audio_new ()
#define	AGS_FX_PEAK_AUDIO()
#define	AGS_FX_PEAK_AUDIO_CLASS()
#define	AGS_FX_PEAK_AUDIO_GET_CLASS()
#define	AGS_IS_FX_PEAK_AUDIO()
#define	AGS_IS_FX_PEAK_AUDIO_CLASS()
GType	ags_fx_peak_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_PEAK_AUDIO
struct	AgsFxPeakAudio
struct	AgsFxPeakAudioClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudio
        &#x2570;&#x2500;&#x2500; AgsFxPeakAudio
```

Implemented Interfaces

AgsFxPeakAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_peak_audio.h>
```

Description

The AgsFxPeakAudio class provides ports to the effect processor.

Functions

ags_fx_peak_audio_new ()

```
AgsFxPeakAudio~*
ags_fx_peak_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxPeakAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxPeakAudio

Since: 3.3.0

AGS_FX_PEAK_AUDIO()

```
#define AGS_FX_PEAK_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO, AgsFxPeakAudio))
```

AGS_FX_PEAK_AUDIO_CLASS()

```
#define AGS_FX_PEAK_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PEAK_AUDIO, AgsFxPeakAudioClass))
```

AGS_FX_PEAK_AUDIO_GET_CLASS()

```
#define AGS_FX_PEAK_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO, AgsFxPeakAudioClass))
```

AGS_IS_FX_PEAK_AUDIO()

```
#define AGS_IS_FX_PEAK_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO))
```

AGS_IS_FX_PEAK_AUDIO_CLASS()

```
#define AGS_IS_FX_PEAK_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PEAK_AUDIO))
```

ags_fx_peak_audio_get_type ()

```
GType
ags_fx_peak_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_PEAK_AUDIO

```
#define AGS_TYPE_FX_PEAK_AUDIO                 (ags_fx_peak_audio_get_type())
```

struct AgsFxPeakAudio

```
struct AgsFxPeakAudio;
```

struct AgsFxPeakAudioClass

```
struct AgsFxPeakAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

21.2 AgsFxPeakAudioProcessor

AgsFxPeakAudioProcessor — fx peak audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPeakAudioProcessor *	ags_fx_peak_audio_processor_new ()
#define	AGS_FX_PEAK_AUDIO_PROCESSOR()
#define	AGS_FX_PEAK_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_PEAK_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_PEAK_AUDIO_PROCESSOR()
#define	AGS_IS_FX_PEAK_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_peak_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR
struct	AgsFxPeakAudioProcessor
struct	AgsFxPeakAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxPeakAudioProcessor
```

Implemented Interfaces

AgsFxPeakAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_peak_audio_processor.h>
```

Description

The AgsFxPeakAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_peak_audio_processor_new ()

```
AgsFxPeakAudioProcessor~*
ags_fx_peak_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxPeakAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxPeakAudioProcessor

Since: 3.3.0

AGS_FX_PEAK_AUDIO_PROCESSOR()

```
#define AGS_FX_PEAK_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR, AgsFxPeakAudioProcessor))
```

AGS_FX_PEAK_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_PEAK_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR, AgsFxPeakAudioProcessorClass))
```

AGS_FX_PEAK_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_PEAK_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR, AgsFxPeakAudioProcessorClass))
```

AGS_IS_FX_PEAK_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_PEAK_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
     AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR))
```

AGS_IS_FX_PEAK_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_PEAK_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR))
```

ags_fx_peak_audio_processor_get_type ()

```
GType
ags_fx_peak_audio_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR                ( ↩
    ags_fx_peak_audio_processor_get_type())
```

struct AgsFxPeakAudioProcessor

```
struct AgsFxPeakAudioProcessor;
```

struct AgsFxPeakAudioProcessorClass

```
struct AgsFxPeakAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

21.3 AgsFxPeakAudioSignal

AgsFxPeakAudioSignal — fx peak audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPeakAudioSignal *	ags_fx_peak_audio_signal_new ()
#define	AGS_FX_PEAK_AUDIO_SIGNAL()
#define	AGS_FX_PEAK_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_PEAK_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_PEAK_AUDIO_SIGNAL()
#define	AGS_IS_FX_PEAK_AUDIO_SIGNAL_CLASS()
GType	ags_fx_peak_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_PEAK_AUDIO_SIGNAL
struct	AgsFxPeakAudioSignal
struct	AgsFxPeakAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxPeakAudioSignal
```

Implemented Interfaces

AgsFxPeakAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_peak_audio_signal.h>
```

Description

The AgsFxPeakAudioSignal class provides ports to the effect processor.

Functions

ags_fx_peak_audio_signal_new ()

```
AgsFxPeakAudioSignal~*
ags_fx_peak_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxPeakAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxPeakAudioSignal

Since: 3.3.0

AGS_FX_PEAK_AUDIO_SIGNAL()

```
#define AGS_FX_PEAK_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_SIGNAL, AgsFxPeakAudioSignal))
```

AGS_FX_PEAK_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_PEAK_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_SIGNAL, AgsFxPeakAudioSignalClass))
```

AGS_FX_PEAK_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_PEAK_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_SIGNAL, AgsFxPeakAudioSignalClass))
```

AGS_IS_FX_PEAK_AUDIO_SIGNAL()

```
#define AGS_IS_FX_PEAK_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_SIGNAL))
```

AGS_IS_FX_PEAK_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_PEAK_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PEAK_AUDIO_SIGNAL))
```

ags_fx_peak_audio_signal_get_type ()

```
GType
ags_fx_peak_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_PEAK_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_PEAK_AUDIO_SIGNAL                (ags_fx_peak_audio_signal_get_type())
```

struct AgsFxPeakAudioSignal

```
struct AgsFxPeakAudioSignal;
```

struct AgsFxPeakAudioSignalClass

```
struct AgsFxPeakAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

21.4 AgsFxPeakChannel

AgsFxPeakChannel — fx peak channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_PEAK_CHANNEL_INPUT_DATA()
#define	AGS_FX_PEAK_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxPeakChannelInputData *	ags_fx_peak_channel_input_data_alloc ()
void	ags_fx_peak_channel_input_data_free ()
AgsFxPeakChannel *	ags_fx_peak_channel_new ()
#define	AGS_FX_PEAK_CHANNEL()
#define	AGS_FX_PEAK_CHANNEL_CLASS()
#define	AGS_FX_PEAK_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_PEAK_CHANNEL()
#define	AGS_IS_FX_PEAK_CHANNEL_CLASS()
GType	ags_fx_peak_channel_get_type ()

Properties

AgsPort *	peak	Read / Write

Types and Values

struct	AgsFxPeakChannelInputData
#define	AGS_TYPE_FX_PEAK_CHANNEL
struct	AgsFxPeakChannel
struct	AgsFxPeakChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxPeakChannel
```

Implemented Interfaces

AgsFxPeakChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_peak_channel.h>
```

Description

The AgsFxPeakChannel class provides ports to the effect processor.

Functions

AGS_FX_PEAK_CHANNEL_INPUT_DATA()

```
#define AGS_FX_PEAK_CHANNEL_INPUT_DATA(ptr) ((AgsFxPeakChannelInputData *)(ptr))
```

AGS_FX_PEAK_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_PEAK_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxPeakChannelInputData ↩
    *)(ptr))->strct_mutex))
```

ags_fx_peak_channel_input_data_alloc ()

```
AgsFxPeakChannelInputData~*
ags_fx_peak_channel_input_data_alloc ();
```

Allocate AgsFxPeakChannelInputData

Returns

the new AgsFxPeakChannelInputData

Since: 3.3.0

ags_fx_peak_channel_input_data_free ()

```
void
ags_fx_peak_channel_input_data_free (AgsFxPeakChannelInputData *input_data);
```

Free *input_data*.

Parameters

input_data	the AgsFxPeakChannelInputData

Since: 3.3.0

ags_fx_peak_channel_new ()

```
AgsFxPeakChannel~*
ags_fx_peak_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxPeakChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxPeakChannel

Since: 3.3.0

AGS_FX_PEAK_CHANNEL()

```
#define AGS_FX_PEAK_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PEAK_CHANNEL, AgsFxPeakChannel))
```

AGS_FX_PEAK_CHANNEL_CLASS()

```
#define AGS_FX_PEAK_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PEAK_CHANNEL, AgsFxPeakChannelClass))
```

AGS_FX_PEAK_CHANNEL_GET_CLASS()

```
#define AGS_FX_PEAK_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PEAK_CHANNEL, AgsFxPeakChannelClass))
```

AGS_IS_FX_PEAK_CHANNEL()

```
#define AGS_IS_FX_PEAK_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PEAK_CHANNEL))
```

AGS_IS_FX_PEAK_CHANNEL_CLASS()

```
#define AGS_IS_FX_PEAK_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PEAK_CHANNEL))
```

ags_fx_peak_channel_get_type ()

```
GType
ags_fx_peak_channel_get_type ();
```

Types and Values

struct AgsFxPeakChannelInputData

```
struct AgsFxPeakChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  void *buffer;
};
```

AGS_TYPE_FX_PEAK_CHANNEL

```
#define AGS_TYPE_FX_PEAK_CHANNEL                 (ags_fx_peak_channel_get_type())
```

struct AgsFxPeakChannel

```
struct AgsFxPeakChannel;
```

struct AgsFxPeakChannelClass

```
struct AgsFxPeakChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "peak" property

```
  "peak"                      AgsPort~*
```

The beats per minute.

Owner: AgsFxPeakChannel

Flags: Read / Write

Since: 3.3.0

21.5 AgsFxPeakChannelProcessor

AgsFxPeakChannelProcessor — fx peak channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPeakChannelProcessor *	ags_fx_peak_channel_processor_new ()
#define	AGS_FX_PEAK_CHANNEL_PROCESSOR()
#define	AGS_FX_PEAK_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_PEAK_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_PEAK_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_PEAK_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_peak_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR
struct	AgsFxPeakChannelProcessor
struct	AgsFxPeakChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxPeakChannelProcessor
```

Implemented Interfaces

AgsFxPeakChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_peak_channel_processor.h>
```

Description

The AgsFxPeakChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_peak_channel_processor_new ()

```
AgsFxPeakChannelProcessor~*
ags_fx_peak_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxPeakChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxPeakChannelProcessor

Since: 3.3.0

AGS_FX_PEAK_CHANNEL_PROCESSOR()

```
#define AGS_FX_PEAK_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR, AgsFxPeakChannelProcessor))
```

AGS_FX_PEAK_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_PEAK_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR, AgsFxPeakChannelProcessorClass))
```

AGS_FX_PEAK_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_PEAK_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR, AgsFxPeakChannelProcessorClass))
```

AGS_IS_FX_PEAK_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_PEAK_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR))
```

AGS_IS_FX_PEAK_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_PEAK_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR))
```

ags_fx_peak_channel_processor_get_type ()

```
GType
ags_fx_peak_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR                ( ↩
    ags_fx_peak_channel_processor_get_type())
```

struct AgsFxPeakChannelProcessor

```
struct AgsFxPeakChannelProcessor;
```

struct AgsFxPeakChannelProcessorClass

```
struct AgsFxPeakChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

21.6 AgsFxPeakRecycling

AgsFxPeakRecycling — fx peak recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxPeakRecycling *	ags_fx_peak_recycling_new ()
#define	AGS_FX_PEAK_RECYCLING()
#define	AGS_FX_PEAK_RECYCLING_CLASS()
#define	AGS_FX_PEAK_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_PEAK_RECYCLING()
#define	AGS_IS_FX_PEAK_RECYCLING_CLASS()
GType	ags_fx_peak_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_PEAK_RECYCLING
struct	AgsFxPeakRecycling
struct	AgsFxPeakRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxPeakRecycling
```

Implemented Interfaces

AgsFxPeakRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_peak_recycling.h>
```

Description

The AgsFxPeakRecycling class provides ports to the effect processor.

Functions

ags_fx_peak_recycling_new ()

```
AgsFxPeakRecycling~*
ags_fx_peak_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxPeakRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxPeakRecycling

Since: 3.3.0

AGS_FX_PEAK_RECYCLING()

```
#define AGS_FX_PEAK_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_PEAK_RECYCLING, AgsFxPeakRecycling))
```

AGS_FX_PEAK_RECYCLING_CLASS()

```
#define AGS_FX_PEAK_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_PEAK_RECYCLING, AgsFxPeakRecyclingClass))
```

AGS_FX_PEAK_RECYCLING_GET_CLASS()

```
#define AGS_FX_PEAK_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_PEAK_RECYCLING, AgsFxPeakRecyclingClass))
```

AGS_IS_FX_PEAK_RECYCLING()

```
#define AGS_IS_FX_PEAK_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_PEAK_RECYCLING))
```

AGS_IS_FX_PEAK_RECYCLING_CLASS()

```
#define AGS_IS_FX_PEAK_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_PEAK_RECYCLING))
```

ags_fx_peak_recycling_get_type ()

```
GType
ags_fx_peak_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_PEAK_RECYCLING

```
#define AGS_TYPE_FX_PEAK_RECYCLING                (ags_fx_peak_recycling_get_type())
```

struct AgsFxPeakRecycling

```
struct AgsFxPeakRecycling;
```

struct AgsFxPeakRecyclingClass

```
struct AgsFxPeakRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

21.7 AgsFxAnalyseAudio

AgsFxAnalyseAudio — fx analyse audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxAnalyseAudio *	ags_fx_analyse_audio_new ()
#define	AGS_FX_ANALYSE_AUDIO()
#define	AGS_FX_ANALYSE_AUDIO_CLASS()
#define	AGS_FX_ANALYSE_AUDIO_GET_CLASS()
#define	AGS_IS_FX_ANALYSE_AUDIO()
#define	AGS_IS_FX_ANALYSE_AUDIO_CLASS()
GType	ags_fx_analyse_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_ANALYSE_AUDIO
struct	AgsFxAnalyseAudio
struct	AgsFxAnalyseAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxAnalyseAudio
```

Implemented Interfaces

AgsFxAnalyseAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_analyse_audio.h>
```

Description

The AgsFxAnalyseAudio class provides ports to the effect processor.

Functions

ags_fx_analyse_audio_new ()

```
AgsFxAnalyseAudio~*
ags_fx_analyse_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxAnalyseAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxAnalyseAudio

Since: 3.3.0

AGS_FX_ANALYSE_AUDIO()

```
#define AGS_FX_ANALYSE_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_ANALYSE_AUDIO, AgsFxAnalyseAudio))
```

AGS_FX_ANALYSE_AUDIO_CLASS()

```
#define AGS_FX_ANALYSE_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_ANALYSE_AUDIO, AgsFxAnalyseAudioClass))
```

AGS_FX_ANALYSE_AUDIO_GET_CLASS()

```
#define AGS_FX_ANALYSE_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_ANALYSE_AUDIO, AgsFxAnalyseAudioClass))
```

AGS_IS_FX_ANALYSE_AUDIO()

```
#define AGS_IS_FX_ANALYSE_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_ANALYSE_AUDIO))
```

AGS_IS_FX_ANALYSE_AUDIO_CLASS()

```
#define AGS_IS_FX_ANALYSE_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_ANALYSE_AUDIO))
```

ags_fx_analyse_audio_get_type ()

```
GType
ags_fx_analyse_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_ANALYSE_AUDIO

```
#define AGS_TYPE_FX_ANALYSE_AUDIO                (ags_fx_analyse_audio_get_type())
```

struct AgsFxAnalyseAudio

```
struct AgsFxAnalyseAudio;
```

struct AgsFxAnalyseAudioClass

```
struct AgsFxAnalyseAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

21.8 AgsFxAnalyseAudioProcessor

AgsFxAnalyseAudioProcessor — fx analyse audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxAnalyseAudioProcessor *	ags_fx_analyse_audio_processor_new ()
#define	AGS_FX_ANALYSE_AUDIO_PROCESSOR()
#define	AGS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_ANALYSE_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR()
#define	AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_analyse_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR
struct	AgsFxAnalyseAudioProcessor
struct	AgsFxAnalyseAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxAnalyseAudioProcessor
```

Implemented Interfaces

AgsFxAnalyseAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_analyse_audio_processor.h>
```

Description

The AgsFxAnalyseAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_analyse_audio_processor_new ()

```
AgsFxAnalyseAudioProcessor~*
ags_fx_analyse_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxAnalyseAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxAnalyseAudioProcessor

Since: 3.3.0

AGS_FX_ANALYSE_AUDIO_PROCESSOR()

```
#define AGS_FX_ANALYSE_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR, AgsFxAnalyseAudioProcessor))
```

AGS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class) ↩
    , AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR, AgsFxAnalyseAudioProcessorClass))
```

AGS_FX_ANALYSE_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_ANALYSE_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj ↩
    ), AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR, AgsFxAnalyseAudioProcessorClass))
```

AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR))
```

AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR))
```

ags_fx_analyse_audio_processor_get_type ()

```
GType
ags_fx_analyse_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR                ( ↩
    ags_fx_analyse_audio_processor_get_type())
```

struct AgsFxAnalyseAudioProcessor

```
struct AgsFxAnalyseAudioProcessor;
```

struct AgsFxAnalyseAudioProcessorClass

```
struct AgsFxAnalyseAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

21.9 AgsFxAnalyseAudioSignal

AgsFxAnalyseAudioSignal — fx analyse audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxAnalyseAudioSignal *	ags_fx_analyse_audio_signal_new ()
#define	AGS_FX_ANALYSE_AUDIO_SIGNAL()
#define	AGS_FX_ANALYSE_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_ANALYSE_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_ANALYSE_AUDIO_SIGNAL()
#define	AGS_IS_FX_ANALYSE_AUDIO_SIGNAL_CLASS()
GType	ags_fx_analyse_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL
struct	AgsFxAnalyseAudioSignal
struct	AgsFxAnalyseAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxAnalyseAudioSignal
```

Implemented Interfaces

AgsFxAnalyseAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_analyse_audio_signal.h>
```

Description

The AgsFxAnalyseAudioSignal class provides ports to the effect processor.

Functions

ags_fx_analyse_audio_signal_new ()

```
AgsFxAnalyseAudioSignal~*
ags_fx_analyse_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxAnalyseAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxAnalyseAudioSignal

Since: 3.3.0

AGS_FX_ANALYSE_AUDIO_SIGNAL()

```
#define AGS_FX_ANALYSE_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL, AgsFxAnalyseAudioSignal))
```

AGS_FX_ANALYSE_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_ANALYSE_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
    AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL, AgsFxAnalyseAudioSignalClass))
```

AGS_FX_ANALYSE_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_ANALYSE_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
    AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL, AgsFxAnalyseAudioSignalClass))
```

AGS_IS_FX_ANALYSE_AUDIO_SIGNAL()

```
#define AGS_IS_FX_ANALYSE_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
     AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL))
```

AGS_IS_FX_ANALYSE_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_ANALYSE_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
    AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL))
```

ags_fx_analyse_audio_signal_get_type ()

```
GType
ags_fx_analyse_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL                ( ↩
    ags_fx_analyse_audio_signal_get_type())
```

struct AgsFxAnalyseAudioSignal

```
struct AgsFxAnalyseAudioSignal;
```

struct AgsFxAnalyseAudioSignalClass

```
struct AgsFxAnalyseAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

21.10 AgsFxAnalyseChannel

AgsFxAnalyseChannel — fx analyse channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_ANALYSE_CHANNEL_INPUT_DATA()
#define	AGS_FX_ANALYSE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxAnalyseChannelInputData *	ags_fx_analyse_channel_input_data_alloc ()
void	ags_fx_analyse_channel_input_data_free ()
GRecMutex *	ags_fx_analyse_channel_input_data_get_strct_mutex ()
gpointer	ags_fx_analyse_channel_input_get_parent ()
gpointer	ags_fx_analyse_channel_input_get_in ()
gpointer	ags_fx_analyse_channel_input_get_out ()
AgsFxAnalyseChannelInputData *	ags_fx_analyse_channel_get_input_data ()
AgsFxAnalyseChannel *	ags_fx_analyse_channel_new ()
#define	AGS_FX_ANALYSE_CHANNEL()
#define	AGS_FX_ANALYSE_CHANNEL_CLASS()
#define	AGS_FX_ANALYSE_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_ANALYSE_CHANNEL()
#define	AGS_IS_FX_ANALYSE_CHANNEL_CLASS()
GType	ags_fx_analyse_channel_get_type ()

Properties

AgsPort *	frequency	Read / Write
AgsPort *	magnitude	Read / Write

Types and Values

struct	AgsFxAnalyseChannelInputData
#define	AGS_TYPE_FX_ANALYSE_CHANNEL
struct	AgsFxAnalyseChannel
struct	AgsFxAnalyseChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxAnalyseChannel
```

Implemented Interfaces

AgsFxAnalyseChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_analyse_channel.h>
```

Description

The AgsFxAnalyseChannel class provides ports to the effect processor.

Functions

AGS_FX_ANALYSE_CHANNEL_INPUT_DATA()

```
#define AGS_FX_ANALYSE_CHANNEL_INPUT_DATA(ptr) ((AgsFxAnalyseChannelInputData *)(ptr))
```

AGS_FX_ANALYSE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_ANALYSE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&((( ↩
    AgsFxAnalyseChannelInputData *)(ptr))->strct_mutex))
```

ags_fx_analyse_channel_input_data_alloc ()

```
AgsFxAnalyseChannelInputData~*
ags_fx_analyse_channel_input_data_alloc
                                ();
```

Allocate AgsFxAnalyseChannelInputData

Returns

the new AgsFxAnalyseChannelInputData.

[type gpointer][transfer full]

Since: 3.3.0

ags_fx_analyse_channel_input_data_free ()

```
void
ags_fx_analyse_channel_input_data_free
                                (AgsFxAnalyseChannelInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxAnalyseChannelInputData.	*[type gpointer][transfer full]*

Since: 3.3.0

ags_fx_analyse_channel_input_data_get_strct_mutex ()

```
GRecMutex~*
ags_fx_analyse_channel_input_data_get_strct_mutex
                                (AgsFxAnalyseChannelInputData *input_data);
```

Get structure mutex.

Parameters

input_data	the AgsFxAnalyseChannelInputData.	*[type gpointer][transfer none]*

Returns

the GRecMutex to lock *input_data* .

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_analyse_channel_input_get_parent ()

```
gpointer
ags_fx_analyse_channel_input_get_parent
                                (AgsFxAnalyseChannelInputData *input_data);
```

Get parent of *input_data* .

Parameters

input_data	the AgsFxAnalyseChannelInputData.	*[type gpointer][transfer none]*

Returns

the parent.

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_analyse_channel_input_get_in ()

```
gpointer
ags_fx_analyse_channel_input_get_in (AgsFxAnalyseChannelInputData *input_data);
```

Get input of *input_data* .

Parameters

input_data	the AgsFxAnalyseChannelInputData.	*[type gpointer][transfer none]*

Returns

the input.

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_analyse_channel_input_get_out ()

```
gpointer
ags_fx_analyse_channel_input_get_out (AgsFxAnalyseChannelInputData *input_data);
```

Get output of *input_data* .

Parameters

input_data	the AgsFxAnalyseChannelInputData.	*[type gpointer][transfer none]*

Returns

the output.

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_analyse_channel_get_input_data ()

```
AgsFxAnalyseChannelInputData~*
ags_fx_analyse_channel_get_input_data (AgsFxAnalyseChannel *fx_analyse_channel,
                                       gint sound_scope);
```

Get input data from *fx_analyse_channel* by *sound_scope* .

Parameters

fx_analyse_channel	the AgsFxAnalyseChannel	
sound_scope	the sound scope	

Returns

the matching AgsFxAnalyseChannelInputData.

[type gpointer][transfer none]

Since: 3.3.0

ags_fx_analyse_channel_new ()

```
AgsFxAnalyseChannel~*
ags_fx_analyse_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxAnalyseChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxAnalyseChannel

Since: 3.3.0

AGS_FX_ANALYSE_CHANNEL()

```
#define AGS_FX_ANALYSE_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_ANALYSE_CHANNEL, AgsFxAnalyseChannel))
```

AGS_FX_ANALYSE_CHANNEL_CLASS()

```
#define AGS_FX_ANALYSE_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_ANALYSE_CHANNEL, AgsFxAnalyseChannelClass))
```

AGS_FX_ANALYSE_CHANNEL_GET_CLASS()

```
#define AGS_FX_ANALYSE_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_ANALYSE_CHANNEL, AgsFxAnalyseChannelClass))
```

AGS_IS_FX_ANALYSE_CHANNEL()

```
#define AGS_IS_FX_ANALYSE_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_ANALYSE_CHANNEL))
```

AGS_IS_FX_ANALYSE_CHANNEL_CLASS()

```
#define AGS_IS_FX_ANALYSE_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_ANALYSE_CHANNEL))
```

ags_fx_analyse_channel_get_type ()

```
GType
ags_fx_analyse_channel_get_type ();
```

Types and Values

struct AgsFxAnalyseChannelInputData

```
struct AgsFxAnalyseChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  fftw_plan plan;
  fftw_complex *comout;

  double *in;
  double *out;
};
```

AGS_TYPE_FX_ANALYSE_CHANNEL

```
#define AGS_TYPE_FX_ANALYSE_CHANNEL                (ags_fx_analyse_channel_get_type())
```

struct AgsFxAnalyseChannel

```
struct AgsFxAnalyseChannel;
```

struct AgsFxAnalyseChannelClass

```
struct AgsFxAnalyseChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "frequency" property

```
  "frequency"                AgsPort~*
```

The beats per minute.

Owner: AgsFxAnalyseChannel

Flags: Read / Write

Since: 3.3.0

The "magnitude" property

```
"magnitude"                AgsPort~*
```

The beats per minute.

Owner: AgsFxAnalyseChannel

Flags: Read / Write

Since: 3.3.0

21.11 AgsFxAnalyseChannelProcessor

AgsFxAnalyseChannelProcessor — fx analyse channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxAnalyseChannelProcessor *	ags_fx_analyse_channel_processor_new ()
#define	AGS_FX_ANALYSE_CHANNEL_PROCESSOR()
#define	AGS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_ANALYSE_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_analyse_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR
struct	AgsFxAnalyseChannelProcessor
struct	AgsFxAnalyseChannelProcessorClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
        &#x2570;&#x2500;&#x2500; AgsFxAnalyseChannelProcessor
```

Implemented Interfaces

AgsFxAnalyseChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_analyse_channel_processor.h>
```

Description

The AgsFxAnalyseChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_analyse_channel_processor_new ()

```
AgsFxAnalyseChannelProcessor~*
ags_fx_analyse_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxAnalyseChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxAnalyseChannelProcessor

Since: 3.3.0

AGS_FX_ANALYSE_CHANNEL_PROCESSOR()

```
#define AGS_FX_ANALYSE_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR, AgsFxAnalyseChannelProcessor))
```

AGS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR, AgsFxAnalyseChannelProcessorClass))
```

AGS_FX_ANALYSE_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_ANALYSE_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR, AgsFxAnalyseChannelProcessorClass))
```

AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR))
```

AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR))
```

ags_fx_analyse_channel_processor_get_type ()

```
GType
ags_fx_analyse_channel_processor_get_type
                                    ();
```

Types and Values

AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR                ( ↩
    ags_fx_analyse_channel_processor_get_type())
```

struct AgsFxAnalyseChannelProcessor

```
struct AgsFxAnalyseChannelProcessor;
```

struct AgsFxAnalyseChannelProcessorClass

```
struct AgsFxAnalyseChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

21.12 AgsFxAnalyseRecycling

AgsFxAnalyseRecycling — fx analyse recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxAnalyseRecycling *	ags_fx_analyse_recycling_new ()
#define	AGS_FX_ANALYSE_RECYCLING()
#define	AGS_FX_ANALYSE_RECYCLING_CLASS()
#define	AGS_FX_ANALYSE_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_ANALYSE_RECYCLING()
#define	AGS_IS_FX_ANALYSE_RECYCLING_CLASS()
GType	ags_fx_analyse_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_ANALYSE_RECYCLING
struct	AgsFxAnalyseRecycling
struct	AgsFxAnalyseRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxAnalyseRecycling
```

Implemented Interfaces

AgsFxAnalyseRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_analyse_recycling.h>
```

Description

The AgsFxAnalyseRecycling class provides ports to the effect processor.

Functions

ags_fx_analyse_recycling_new ()

```
AgsFxAnalyseRecycling~*
ags_fx_analyse_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxAnalyseRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxAnalyseRecycling

Since: 3.3.0

AGS_FX_ANALYSE_RECYCLING()

```
#define AGS_FX_ANALYSE_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_ANALYSE_RECYCLING, AgsFxAnalyseRecycling))
```

AGS_FX_ANALYSE_RECYCLING_CLASS()

```
#define AGS_FX_ANALYSE_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_ANALYSE_RECYCLING, AgsFxAnalyseRecyclingClass))
```

AGS_FX_ANALYSE_RECYCLING_GET_CLASS()

```
#define AGS_FX_ANALYSE_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_ANALYSE_RECYCLING, AgsFxAnalyseRecyclingClass))
```

AGS_IS_FX_ANALYSE_RECYCLING()

```
#define AGS_IS_FX_ANALYSE_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_ANALYSE_RECYCLING))
```

AGS_IS_FX_ANALYSE_RECYCLING_CLASS()

```
#define AGS_IS_FX_ANALYSE_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_ANALYSE_RECYCLING))
```

ags_fx_analyse_recycling_get_type ()

```
GType
ags_fx_analyse_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_ANALYSE_RECYCLING

```
#define AGS_TYPE_FX_ANALYSE_RECYCLING                (ags_fx_analyse_recycling_get_type())
```

struct AgsFxAnalyseRecycling

```
struct AgsFxAnalyseRecycling;
```

struct AgsFxAnalyseRecyclingClass

```
struct AgsFxAnalyseRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

Chapter 22

Fx Engine - volume and envelope

Volume adjustment or mute and envelope of audio data.

22.1 AgsFxVolumeAudio

AgsFxVolumeAudio — fx volume audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxVolumeAudio *	ags_fx_volume_audio_new ()
#define	AGS_FX_VOLUME_AUDIO()
#define	AGS_FX_VOLUME_AUDIO_CLASS()
#define	AGS_FX_VOLUME_AUDIO_GET_CLASS()
#define	AGS_IS_FX_VOLUME_AUDIO()
#define	AGS_IS_FX_VOLUME_AUDIO_CLASS()
GType	ags_fx_volume_audio_get_type ()

Properties

AgsPort *	muted	Read / Write

Types and Values

#define	AGS_TYPE_FX_VOLUME_AUDIO
struct	AgsFxVolumeAudio
struct	AgsFxVolumeAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxVolumeAudio
```

Implemented Interfaces

AgsFxVolumeAudio implements AgsConnectable and AgsMutable.

Includes

```
#include <ags/audio/fx/ags_fx_volume_audio.h>
```

Description

The AgsFxVolumeAudio class provides ports to the effect processor.

Functions

ags_fx_volume_audio_new ()

```
AgsFxVolumeAudio~*
ags_fx_volume_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxVolumeAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxVolumeAudio

Since: 3.3.0

AGS_FX_VOLUME_AUDIO()

```
#define AGS_FX_VOLUME_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO, AgsFxVolumeAudio))
```

AGS_FX_VOLUME_AUDIO_CLASS()

```
#define AGS_FX_VOLUME_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO, AgsFxVolumeAudioClass))
```

AGS_FX_VOLUME_AUDIO_GET_CLASS()

```
#define AGS_FX_VOLUME_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO, AgsFxVolumeAudioClass))
```

AGS_IS_FX_VOLUME_AUDIO()

```
#define AGS_IS_FX_VOLUME_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO))
```

AGS_IS_FX_VOLUME_AUDIO_CLASS()

```
#define AGS_IS_FX_VOLUME_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO))
```

ags_fx_volume_audio_get_type ()

```
GType
ags_fx_volume_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_VOLUME_AUDIO

```
#define AGS_TYPE_FX_VOLUME_AUDIO                (ags_fx_volume_audio_get_type())
```

struct AgsFxVolumeAudio

```
struct AgsFxVolumeAudio;
```

struct AgsFxVolumeAudioClass

```
struct AgsFxVolumeAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

Property Details

The "muted" property

```
  "muted"                     AgsPort~*
```

The muted port.

Owner: AgsFxVolumeAudio

Flags: Read / Write

Since: 3.3.0

22.2 AgsFxVolumeAudioProcessor

AgsFxVolumeAudioProcessor — fx volume audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxVolumeAudioProcessor *	ags_fx_volume_audio_processor_new ()
#define	AGS_FX_VOLUME_AUDIO_PROCESSOR()
#define	AGS_FX_VOLUME_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_VOLUME_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_VOLUME_AUDIO_PROCESSOR()
#define	AGS_IS_FX_VOLUME_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_volume_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR
struct	AgsFxVolumeAudioProcessor
struct	AgsFxVolumeAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxVolumeAudioProcessor
```

Implemented Interfaces

AgsFxVolumeAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_volume_audio_processor.h>
```

Description

The AgsFxVolumeAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_volume_audio_processor_new ()

```
AgsFxVolumeAudioProcessor~*
ags_fx_volume_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxVolumeAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxVolumeAudioProcessor

Since: 3.3.0

AGS_FX_VOLUME_AUDIO_PROCESSOR()

```
#define AGS_FX_VOLUME_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR, AgsFxVolumeAudioProcessor))
```

AGS_FX_VOLUME_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_VOLUME_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR, AgsFxVolumeAudioProcessorClass))
```

AGS_FX_VOLUME_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_VOLUME_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR, AgsFxVolumeAudioProcessorClass))
```

AGS_IS_FX_VOLUME_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_VOLUME_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR))
```

AGS_IS_FX_VOLUME_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_VOLUME_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR))
```

ags_fx_volume_audio_processor_get_type ()

```
GType
ags_fx_volume_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR                ( ↩
    ags_fx_volume_audio_processor_get_type())
```

struct AgsFxVolumeAudioProcessor

```
struct AgsFxVolumeAudioProcessor;
```

struct AgsFxVolumeAudioProcessorClass

```
struct AgsFxVolumeAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

22.3 AgsFxVolumeAudioSignal

AgsFxVolumeAudioSignal — fx volume audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxVolumeAudioSignal *	ags_fx_volume_audio_signal_new ()
#define	AGS_FX_VOLUME_AUDIO_SIGNAL()
#define	AGS_FX_VOLUME_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_VOLUME_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_VOLUME_AUDIO_SIGNAL()
#define	AGS_IS_FX_VOLUME_AUDIO_SIGNAL_CLASS()
GType	ags_fx_volume_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL
struct	AgsFxVolumeAudioSignal
struct	AgsFxVolumeAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxVolumeAudioSignal
```

Implemented Interfaces

AgsFxVolumeAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_volume_audio_signal.h>
```

Description

The AgsFxVolumeAudioSignal class provides ports to the effect processor.

Functions

ags_fx_volume_audio_signal_new ()

```
AgsFxVolumeAudioSignal~*
ags_fx_volume_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxVolumeAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxVolumeAudioSignal

Since: 3.3.0

AGS_FX_VOLUME_AUDIO_SIGNAL()

```
#define AGS_FX_VOLUME_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL, AgsFxVolumeAudioSignal))
```

AGS_FX_VOLUME_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_VOLUME_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL, AgsFxVolumeAudioSignalClass))
```

AGS_FX_VOLUME_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_VOLUME_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL, AgsFxVolumeAudioSignalClass))
```

AGS_IS_FX_VOLUME_AUDIO_SIGNAL()

```
#define AGS_IS_FX_VOLUME_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL))
```

AGS_IS_FX_VOLUME_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_VOLUME_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL))
```

ags_fx_volume_audio_signal_get_type ()

```
GType
ags_fx_volume_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL                (ags_fx_volume_audio_signal_get_type ↩
    ())
```

struct AgsFxVolumeAudioSignal

```
struct AgsFxVolumeAudioSignal;
```

struct AgsFxVolumeAudioSignalClass

```
struct AgsFxVolumeAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

22.4 AgsFxVolumeChannel

AgsFxVolumeChannel — fx volume channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxVolumeChannel *	ags_fx_volume_channel_new ()
#define	AGS_FX_VOLUME_CHANNEL()
#define	AGS_FX_VOLUME_CHANNEL_CLASS()
#define	AGS_FX_VOLUME_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_VOLUME_CHANNEL()
#define	AGS_IS_FX_VOLUME_CHANNEL_CLASS()
GType	ags_fx_volume_channel_get_type ()

Properties

AgsPort *	muted	Read / Write
AgsPort *	volume	Read / Write

Types and Values

#define	AGS_TYPE_FX_VOLUME_CHANNEL
struct	AgsFxVolumeChannel
struct	AgsFxVolumeChannelClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannel
        &#x2570;&#x2500;&#x2500; AgsFxVolumeChannel
```

Implemented Interfaces

AgsFxVolumeChannel implements AgsConnectable and AgsMutable.

Includes

```
#include <ags/audio/fx/ags_fx_volume_channel.h>
```

Description

The AgsFxVolumeChannel class provides ports to the effect processor.

Functions

ags_fx_volume_channel_new ()

```
AgsFxVolumeChannel~*
ags_fx_volume_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxVolumeChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxVolumeChannel

Since: 3.3.0

AGS_FX_VOLUME_CHANNEL()

```
#define AGS_FX_VOLUME_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_VOLUME_CHANNEL, AgsFxVolumeChannel))
```

AGS_FX_VOLUME_CHANNEL_CLASS()

```
#define AGS_FX_VOLUME_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_VOLUME_CHANNEL, AgsFxVolumeChannelClass))
```

AGS_FX_VOLUME_CHANNEL_GET_CLASS()

```
#define AGS_FX_VOLUME_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_VOLUME_CHANNEL, AgsFxVolumeChannelClass))
```

AGS_IS_FX_VOLUME_CHANNEL()

```
#define AGS_IS_FX_VOLUME_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_VOLUME_CHANNEL))
```

AGS_IS_FX_VOLUME_CHANNEL_CLASS()

```
#define AGS_IS_FX_VOLUME_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_VOLUME_CHANNEL))
```

ags_fx_volume_channel_get_type ()

```
GType
ags_fx_volume_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_VOLUME_CHANNEL

```
#define AGS_TYPE_FX_VOLUME_CHANNEL                (ags_fx_volume_channel_get_type())
```

struct AgsFxVolumeChannel

```
struct AgsFxVolumeChannel;
```

struct AgsFxVolumeChannelClass

```
struct AgsFxVolumeChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "muted" property

```
"muted"                    AgsPort~*
```

The muted port.

Owner: AgsFxVolumeChannel

Flags: Read / Write

Since: 3.3.0

The "volume" property

```
"volume"                   AgsPort~*
```

The volume port.

Owner: AgsFxVolumeChannel

Flags: Read / Write

Since: 3.3.0

22.5 AgsFxVolumeChannelProcessor

AgsFxVolumeChannelProcessor — fx volume channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxVolumeChannelProcessor *	ags_fx_volume_channel_processor_new ()
#define	AGS_FX_VOLUME_CHANNEL_PROCESSOR()
#define	AGS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_VOLUME_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_volume_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR
struct	AgsFxVolumeChannelProcessor
struct	AgsFxVolumeChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxVolumeChannelProcessor
```

Implemented Interfaces

AgsFxVolumeChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_volume_channel_processor.h>
```

Description

The AgsFxVolumeChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_volume_channel_processor_new ()

```
AgsFxVolumeChannelProcessor~*
ags_fx_volume_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxVolumeChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxVolumeChannelProcessor

Since: 3.3.0

AGS_FX_VOLUME_CHANNEL_PROCESSOR()

```
#define AGS_FX_VOLUME_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR, AgsFxVolumeChannelProcessor))
```

AGS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR, AgsFxVolumeChannelProcessorClass))
```

AGS_FX_VOLUME_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_VOLUME_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR, AgsFxVolumeChannelProcessorClass))
```

AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR))
```

AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR))
```

ags_fx_volume_channel_processor_get_type ()

```
GType
ags_fx_volume_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR                ( ↩
    ags_fx_volume_channel_processor_get_type())
```

struct AgsFxVolumeChannelProcessor

```
struct AgsFxVolumeChannelProcessor;
```

struct AgsFxVolumeChannelProcessorClass

```
struct AgsFxVolumeChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

22.6 AgsFxVolumeRecycling

AgsFxVolumeRecycling — fx volume recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxVolumeRecycling *	ags_fx_volume_recycling_new ()
#define	AGS_FX_VOLUME_RECYCLING()
#define	AGS_FX_VOLUME_RECYCLING_CLASS()
#define	AGS_FX_VOLUME_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_VOLUME_RECYCLING()
#define	AGS_IS_FX_VOLUME_RECYCLING_CLASS()
GType	ags_fx_volume_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_VOLUME_RECYCLING
struct	AgsFxVolumeRecycling
struct	AgsFxVolumeRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxVolumeRecycling
```

Implemented Interfaces

AgsFxVolumeRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_volume_recycling.h>
```

Description

The AgsFxVolumeRecycling class provides ports to the effect processor.

Functions

ags_fx_volume_recycling_new ()

```
AgsFxVolumeRecycling~*
ags_fx_volume_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxVolumeRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxVolumeRecycling

Since: 3.3.0

AGS_FX_VOLUME_RECYCLING()

```
#define AGS_FX_VOLUME_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_VOLUME_RECYCLING, AgsFxVolumeRecycling))
```

AGS_FX_VOLUME_RECYCLING_CLASS()

```
#define AGS_FX_VOLUME_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_VOLUME_RECYCLING, AgsFxVolumeRecyclingClass))
```

AGS_FX_VOLUME_RECYCLING_GET_CLASS()

```
#define AGS_FX_VOLUME_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_VOLUME_RECYCLING, AgsFxVolumeRecyclingClass))
```

AGS_IS_FX_VOLUME_RECYCLING()

```
#define AGS_IS_FX_VOLUME_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_VOLUME_RECYCLING))
```

AGS_IS_FX_VOLUME_RECYCLING_CLASS()

```
#define AGS_IS_FX_VOLUME_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_VOLUME_RECYCLING))
```

ags_fx_volume_recycling_get_type ()

```
GType
ags_fx_volume_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_VOLUME_RECYCLING

```
#define AGS_TYPE_FX_VOLUME_RECYCLING                (ags_fx_volume_recycling_get_type())
```

struct AgsFxVolumeRecycling

```
struct AgsFxVolumeRecycling;
```

struct AgsFxVolumeRecyclingClass

```
struct AgsFxVolumeRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

22.7 AgsFxLowPassAudio

AgsFxLowPassAudio — fx low pass audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLowPassAudio *	ags_fx_low_pass_audio_new ()
#define	AGS_FX_LOW_PASS_AUDIO()
#define	AGS_FX_LOW_PASS_AUDIO_CLASS()
#define	AGS_FX_LOW_PASS_AUDIO_GET_CLASS()
#define	AGS_IS_FX_LOW_PASS_AUDIO()
#define	AGS_IS_FX_LOW_PASS_AUDIO_CLASS()
GType	ags_fx_low_pass_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_LOW_PASS_AUDIO
struct	AgsFxLowPassAudio
struct	AgsFxLowPassAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxLowPassAudio
```

Implemented Interfaces

AgsFxLowPassAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_low_pass_audio.h>
```

Description

The AgsFxLowPassAudio class provides ports to the effect processor.

Functions

ags_fx_low_pass_audio_new ()

```
AgsFxLowPassAudio~*
ags_fx_low_pass_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxLowPassAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxLowPassAudio

Since: 3.13.0

AGS_FX_LOW_PASS_AUDIO()

```
#define AGS_FX_LOW_PASS_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_AUDIO, AgsFxLowPassAudio))
```

AGS_FX_LOW_PASS_AUDIO_CLASS()

```
#define AGS_FX_LOW_PASS_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LOW_PASS_AUDIO, AgsFxLowPassAudioClass))
```

AGS_FX_LOW_PASS_AUDIO_GET_CLASS()

```
#define AGS_FX_LOW_PASS_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_AUDIO, AgsFxLowPassAudioClass))
```

AGS_IS_FX_LOW_PASS_AUDIO()

```
#define AGS_IS_FX_LOW_PASS_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_AUDIO))
```

AGS_IS_FX_LOW_PASS_AUDIO_CLASS()

```
#define AGS_IS_FX_LOW_PASS_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LOW_PASS_AUDIO))
```

ags_fx_low_pass_audio_get_type ()

```
GType
ags_fx_low_pass_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_LOW_PASS_AUDIO

```
#define AGS_TYPE_FX_LOW_PASS_AUDIO                (ags_fx_low_pass_audio_get_type())
```

struct AgsFxLowPassAudio

```
struct AgsFxLowPassAudio;
```

struct AgsFxLowPassAudioClass

```
struct AgsFxLowPassAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

22.8 AgsFxLowPassAudioProcessor

AgsFxLowPassAudioProcessor — fx low pass audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLowPassAudioProcessor *	ags_fx_low_pass_audio_processor_new ()
#define	AGS_FX_LOW_PASS_AUDIO_PROCESSOR()
#define	AGS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_LOW_PASS_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR()
#define	AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_low_pass_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR
struct	AgsFxLowPassAudioProcessor
struct	AgsFxLowPassAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxLowPassAudioProcessor
```

Implemented Interfaces

AgsFxLowPassAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_low_pass_audio_processor.h>
```

Description

The AgsFxLowPassAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_low_pass_audio_processor_new ()

```
AgsFxLowPassAudioProcessor~*
ags_fx_low_pass_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxLowPassAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxLowPassAudioProcessor

Since: 3.13.0

AGS_FX_LOW_PASS_AUDIO_PROCESSOR()

```
#define AGS_FX_LOW_PASS_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR, AgsFxLowPassAudioProcessor))
```

AGS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR, AgsFxLowPassAudioProcessorClass))
```

AGS_FX_LOW_PASS_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LOW_PASS_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR, AgsFxLowPassAudioProcessorClass))
```

AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR))
```

AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR))
```

ags_fx_low_pass_audio_processor_get_type ()

```
GType
ags_fx_low_pass_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR                ( ↩
    ags_fx_low_pass_audio_processor_get_type())
```

struct AgsFxLowPassAudioProcessor

```
struct AgsFxLowPassAudioProcessor;
```

struct AgsFxLowPassAudioProcessorClass

```
struct AgsFxLowPassAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

22.9 AgsFxLowPassAudioSignal

AgsFxLowPassAudioSignal — fx low pass audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLowPassAudioSignal *	ags_fx_low_pass_audio_signal_new ()
#define	AGS_FX_LOW_PASS_AUDIO_SIGNAL()
#define	AGS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_LOW_PASS_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL()
#define	AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS()
GType	ags_fx_low_pass_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL
struct	AgsFxLowPassAudioSignal
struct	AgsFxLowPassAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxLowPassAudioSignal
```

Implemented Interfaces

AgsFxLowPassAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_low_pass_audio_signal.h>
```

Description

The AgsFxLowPassAudioSignal class provides ports to the effect processor.

Functions

ags_fx_low_pass_audio_signal_new ()

```
AgsFxLowPassAudioSignal~*
ags_fx_low_pass_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxLowPassAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxLowPassAudioSignal

Since: 3.13.0

AGS_FX_LOW_PASS_AUDIO_SIGNAL()

```
#define AGS_FX_LOW_PASS_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL, AgsFxLowPassAudioSignal))
```

AGS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL, AgsFxLowPassAudioSignalClass))
```

AGS_FX_LOW_PASS_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_LOW_PASS_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL, AgsFxLowPassAudioSignalClass))
```

AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL()

```
#define AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL))
```

AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL))
```

ags_fx_low_pass_audio_signal_get_type ()

```
GType
ags_fx_low_pass_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL                ( ↩
    ags_fx_low_pass_audio_signal_get_type())
```

struct AgsFxLowPassAudioSignal

```
struct AgsFxLowPassAudioSignal;
```

struct AgsFxLowPassAudioSignalClass

```
struct AgsFxLowPassAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

22.10 AgsFxLowPassChannel

AgsFxLowPassChannel — fx low pass channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLowPassChannel *	ags_fx_low_pass_channel_new ()
#define	AGS_FX_LOW_PASS_CHANNEL()
#define	AGS_FX_LOW_PASS_CHANNEL_CLASS()
#define	AGS_FX_LOW_PASS_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_LOW_PASS_CHANNEL()
#define	AGS_IS_FX_LOW_PASS_CHANNEL_CLASS()
GType	ags_fx_low_pass_channel_get_type ()

Properties

AgsPort *	enabled	Read / Write
AgsPort *	filter-gain	Read / Write
AgsPort *	q-lin	Read / Write

Types and Values

#define	AGS_TYPE_FX_LOW_PASS_CHANNEL
struct	AgsFxLowPassChannel
struct	AgsFxLowPassChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxLowPassChannel
```

Implemented Interfaces

AgsFxLowPassChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_low_pass_channel.h>
```

Description

The AgsFxLowPassChannel class provides ports to the effect processor.

Functions

ags_fx_low_pass_channel_new ()

```
AgsFxLowPassChannel~*
ags_fx_low_pass_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxLowPassChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxLowPassChannel

Since: 3.13.0

AGS_FX_LOW_PASS_CHANNEL()

```
#define AGS_FX_LOW_PASS_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_CHANNEL, AgsFxLowPassChannel))
```

AGS_FX_LOW_PASS_CHANNEL_CLASS()

```
#define AGS_FX_LOW_PASS_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LOW_PASS_CHANNEL, AgsFxLowPassChannelClass))
```

AGS_FX_LOW_PASS_CHANNEL_GET_CLASS()

```
#define AGS_FX_LOW_PASS_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_CHANNEL, AgsFxLowPassChannelClass))
```

AGS_IS_FX_LOW_PASS_CHANNEL()

```
#define AGS_IS_FX_LOW_PASS_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_CHANNEL))
```

AGS_IS_FX_LOW_PASS_CHANNEL_CLASS()

```
#define AGS_IS_FX_LOW_PASS_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LOW_PASS_CHANNEL))
```

ags_fx_low_pass_channel_get_type ()

```
GType
ags_fx_low_pass_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_LOW_PASS_CHANNEL

```
#define AGS_TYPE_FX_LOW_PASS_CHANNEL                 (ags_fx_low_pass_channel_get_type())
```

struct AgsFxLowPassChannel

```
struct AgsFxLowPassChannel;
```

struct AgsFxLowPassChannelClass

```
struct AgsFxLowPassChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "enabled" property

```
  "enabled"                  AgsPort~*
```

The enabled port.

Owner: AgsFxLowPassChannel

Flags: Read / Write

Since: 3.14.0

The "filter-gain" property

```
  "filter-gain"              AgsPort~*
```

The filter gain port.

Owner: AgsFxLowPassChannel

Flags: Read / Write

Since: 3.13.0

The "q-lin" property

```
"q-lin"                    AgsPort~*
```

The q-lin port.

Owner: AgsFxLowPassChannel

Flags: Read / Write

Since: 3.13.0

22.11 AgsFxLowPassChannelProcessor

AgsFxLowPassChannelProcessor — fx low pass channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLowPassChannelProcessor *	ags_fx_low_pass_channel_processor_new ()
#define	AGS_FX_LOW_PASS_CHANNEL_PROCESSOR()
#define	AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_low_pass_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR
struct	AgsFxLowPassChannelProcessor
struct	AgsFxLowPassChannelProcessorClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
        &#x2570;&#x2500;&#x2500; AgsFxLowPassChannelProcessor
```

Implemented Interfaces

AgsFxLowPassChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_low_pass_channel_processor.h>
```

Description

The AgsFxLowPassChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_low_pass_channel_processor_new ()

```
AgsFxLowPassChannelProcessor~*
ags_fx_low_pass_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxLowPassChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxLowPassChannelProcessor

Since: 3.13.0

AGS_FX_LOW_PASS_CHANNEL_PROCESSOR()

```
#define AGS_FX_LOW_PASS_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR, AgsFxLowPassChannelProcessor))
```

AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR, AgsFxLowPassChannelProcessorClass))
```

AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR, AgsFxLowPassChannelProcessorClass))
```

AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR))
```

AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR))
```

ags_fx_low_pass_channel_processor_get_type ()

```
GType
ags_fx_low_pass_channel_processor_get_type
                                    ();
```

Types and Values

AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR                ( ↩
    ags_fx_low_pass_channel_processor_get_type())
```

struct AgsFxLowPassChannelProcessor

```
struct AgsFxLowPassChannelProcessor;
```

struct AgsFxLowPassChannelProcessorClass

```
struct AgsFxLowPassChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

22.12 AgsFxLowPassRecycling

AgsFxLowPassRecycling — fx low pass recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLowPassRecycling *	ags_fx_low_pass_recycling_new ()
#define	AGS_FX_LOW_PASS_RECYCLING()
#define	AGS_FX_LOW_PASS_RECYCLING_CLASS()
#define	AGS_FX_LOW_PASS_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_LOW_PASS_RECYCLING()
#define	AGS_IS_FX_LOW_PASS_RECYCLING_CLASS()
GType	ags_fx_low_pass_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_LOW_PASS_RECYCLING
struct	AgsFxLowPassRecycling
struct	AgsFxLowPassRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxLowPassRecycling
```

Implemented Interfaces

AgsFxLowPassRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_low_pass_recycling.h>
```

Description

The AgsFxLowPassRecycling class provides ports to the effect processor.

Functions

ags_fx_low_pass_recycling_new ()

```
AgsFxLowPassRecycling~*
ags_fx_low_pass_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxLowPassRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxLowPassRecycling

Since: 3.13.0

AGS_FX_LOW_PASS_RECYCLING()

```
#define AGS_FX_LOW_PASS_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_RECYCLING, AgsFxLowPassRecycling))
```

AGS_FX_LOW_PASS_RECYCLING_CLASS()

```
#define AGS_FX_LOW_PASS_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LOW_PASS_RECYCLING, AgsFxLowPassRecyclingClass))
```

AGS_FX_LOW_PASS_RECYCLING_GET_CLASS()

```
#define AGS_FX_LOW_PASS_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_RECYCLING, AgsFxLowPassRecyclingClass))
```

AGS_IS_FX_LOW_PASS_RECYCLING()

```
#define AGS_IS_FX_LOW_PASS_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LOW_PASS_RECYCLING))
```

AGS_IS_FX_LOW_PASS_RECYCLING_CLASS()

```
#define AGS_IS_FX_LOW_PASS_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LOW_PASS_RECYCLING))
```

ags_fx_low_pass_recycling_get_type ()

```
GType
ags_fx_low_pass_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_LOW_PASS_RECYCLING

```
#define AGS_TYPE_FX_LOW_PASS_RECYCLING                (ags_fx_low_pass_recycling_get_type() ↩
    )
```

struct AgsFxLowPassRecycling

```
struct AgsFxLowPassRecycling;
```

struct AgsFxLowPassRecyclingClass

```
struct AgsFxLowPassRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

22.13 AgsFxHighPassAudio

AgsFxHighPassAudio — fx high pass audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxHighPassAudio *	ags_fx_high_pass_audio_new ()
#define	AGS_FX_HIGH_PASS_AUDIO()
#define	AGS_FX_HIGH_PASS_AUDIO_CLASS()
#define	AGS_FX_HIGH_PASS_AUDIO_GET_CLASS()
#define	AGS_IS_FX_HIGH_PASS_AUDIO()
#define	AGS_IS_FX_HIGH_PASS_AUDIO_CLASS()
GType	ags_fx_high_pass_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_HIGH_PASS_AUDIO
struct	AgsFxHighPassAudio
struct	AgsFxHighPassAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxHighPassAudio
```

Implemented Interfaces

AgsFxHighPassAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_high_pass_audio.h>
```

Description

The AgsFxHighPassAudio class provides ports to the effect processor.

Functions

ags_fx_high_pass_audio_new ()

```
AgsFxHighPassAudio~*
ags_fx_high_pass_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxHighPassAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxHighPassAudio

Since: 3.13.0

AGS_FX_HIGH_PASS_AUDIO()

```
#define AGS_FX_HIGH_PASS_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_AUDIO, AgsFxHighPassAudio))
```

AGS_FX_HIGH_PASS_AUDIO_CLASS()

```
#define AGS_FX_HIGH_PASS_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_HIGH_PASS_AUDIO, AgsFxHighPassAudioClass))
```

AGS_FX_HIGH_PASS_AUDIO_GET_CLASS()

```
#define AGS_FX_HIGH_PASS_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_AUDIO, AgsFxHighPassAudioClass))
```

AGS_IS_FX_HIGH_PASS_AUDIO()

```
#define AGS_IS_FX_HIGH_PASS_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_AUDIO))
```

AGS_IS_FX_HIGH_PASS_AUDIO_CLASS()

```
#define AGS_IS_FX_HIGH_PASS_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_HIGH_PASS_AUDIO))
```

ags_fx_high_pass_audio_get_type ()

```
GType
ags_fx_high_pass_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_HIGH_PASS_AUDIO

```
#define AGS_TYPE_FX_HIGH_PASS_AUDIO                (ags_fx_high_pass_audio_get_type())
```

struct AgsFxHighPassAudio

```
struct AgsFxHighPassAudio;
```

struct AgsFxHighPassAudioClass

```
struct AgsFxHighPassAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

22.14 AgsFxHighPassAudioProcessor

AgsFxHighPassAudioProcessor — fx high pass audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxHighPassAudioProcessor *	ags_fx_high_pass_audio_processor_new ()
#define	AGS_FX_HIGH_PASS_AUDIO_PROCESSOR()
#define	AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR()
#define	AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_high_pass_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR
struct	AgsFxHighPassAudioProcessor
struct	AgsFxHighPassAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxHighPassAudioProcessor
```

Implemented Interfaces

AgsFxHighPassAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_high_pass_audio_processor.h>
```

Description

The AgsFxHighPassAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_high_pass_audio_processor_new ()

```
AgsFxHighPassAudioProcessor~*
ags_fx_high_pass_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxHighPassAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxHighPassAudioProcessor

Since: 3.13.0

AGS_FX_HIGH_PASS_AUDIO_PROCESSOR()

```
#define AGS_FX_HIGH_PASS_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR, AgsFxHighPassAudioProcessor))
```

AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR, AgsFxHighPassAudioProcessorClass))
```

AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR, AgsFxHighPassAudioProcessorClass))
```

AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR))
```

AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR))
```

ags_fx_high_pass_audio_processor_get_type ()

```
GType
ags_fx_high_pass_audio_processor_get_type
                                    ();
```

Types and Values

AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR                ( ↩
    ags_fx_high_pass_audio_processor_get_type())
```

struct AgsFxHighPassAudioProcessor

```
struct AgsFxHighPassAudioProcessor;
```

struct AgsFxHighPassAudioProcessorClass

```
struct AgsFxHighPassAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

22.15 AgsFxHighPassAudioSignal

AgsFxHighPassAudioSignal — fx high pass audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxHighPassAudioSignal *	ags_fx_high_pass_audio_signal_new ()
#define	AGS_FX_HIGH_PASS_AUDIO_SIGNAL()
#define	AGS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_HIGH_PASS_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL()
#define	AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS()
GType	ags_fx_high_pass_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL
struct	AgsFxHighPassAudioSignal
struct	AgsFxHighPassAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxHighPassAudioSignal
```

Implemented Interfaces

AgsFxHighPassAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_high_pass_audio_signal.h>
```

Description

The AgsFxHighPassAudioSignal class provides ports to the effect processor.

Functions

ags_fx_high_pass_audio_signal_new ()

```
AgsFxHighPassAudioSignal~*
ags_fx_high_pass_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxHighPassAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxHighPassAudioSignal

Since: 3.13.0

AGS_FX_HIGH_PASS_AUDIO_SIGNAL()

```
#define AGS_FX_HIGH_PASS_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL, AgsFxHighPassAudioSignal))
```

AGS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL, AgsFxHighPassAudioSignalClass))
```

AGS_FX_HIGH_PASS_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_HIGH_PASS_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL, AgsFxHighPassAudioSignalClass))
```

AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL()

```
#define AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL))
```

AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL))
```

ags_fx_high_pass_audio_signal_get_type ()

```
GType
ags_fx_high_pass_audio_signal_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL                 ( ↩
    ags_fx_high_pass_audio_signal_get_type())
```

struct AgsFxHighPassAudioSignal

```
struct AgsFxHighPassAudioSignal;
```

struct AgsFxHighPassAudioSignalClass

```
struct AgsFxHighPassAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

22.16 AgsFxHighPassChannel

AgsFxHighPassChannel — fx high pass channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxHighPassChannel *	ags_fx_high_pass_channel_new ()
#define	AGS_FX_HIGH_PASS_CHANNEL()
#define	AGS_FX_HIGH_PASS_CHANNEL_CLASS()
#define	AGS_FX_HIGH_PASS_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_HIGH_PASS_CHANNEL()
#define	AGS_IS_FX_HIGH_PASS_CHANNEL_CLASS()
GType	ags_fx_high_pass_channel_get_type ()

Properties

AgsPort *	enabled	Read / Write
AgsPort *	filter-gain	Read / Write
AgsPort *	q-lin	Read / Write

Types and Values

#define	AGS_TYPE_FX_HIGH_PASS_CHANNEL
struct	AgsFxHighPassChannel
struct	AgsFxHighPassChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxHighPassChannel
```

Implemented Interfaces

AgsFxHighPassChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_high_pass_channel.h>
```

Description

The AgsFxHighPassChannel class provides ports to the effect processor.

Functions

ags_fx_high_pass_channel_new ()

```
AgsFxHighPassChannel~*
ags_fx_high_pass_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxHighPassChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxHighPassChannel

Since: 3.13.0

AGS_FX_HIGH_PASS_CHANNEL()

```
#define AGS_FX_HIGH_PASS_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_CHANNEL, AgsFxHighPassChannel))
```

AGS_FX_HIGH_PASS_CHANNEL_CLASS()

```
#define AGS_FX_HIGH_PASS_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_HIGH_PASS_CHANNEL, AgsFxHighPassChannelClass))
```

AGS_FX_HIGH_PASS_CHANNEL_GET_CLASS()

```
#define AGS_FX_HIGH_PASS_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_CHANNEL, AgsFxHighPassChannelClass))
```

AGS_IS_FX_HIGH_PASS_CHANNEL()

```
#define AGS_IS_FX_HIGH_PASS_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_CHANNEL))
```

AGS_IS_FX_HIGH_PASS_CHANNEL_CLASS()

```
#define AGS_IS_FX_HIGH_PASS_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_HIGH_PASS_CHANNEL))
```

ags_fx_high_pass_channel_get_type ()

```
GType
ags_fx_high_pass_channel_get_type ();
```

Types and Values

AGS_TYPE_FX_HIGH_PASS_CHANNEL

```
#define AGS_TYPE_FX_HIGH_PASS_CHANNEL                (ags_fx_high_pass_channel_get_type())
```

struct AgsFxHighPassChannel

```
struct AgsFxHighPassChannel;
```

struct AgsFxHighPassChannelClass

```
struct AgsFxHighPassChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "enabled" property

```
  "enabled"                   AgsPort~*
```

The enabled port.

Owner: AgsFxHighPassChannel

Flags: Read / Write

Since: 3.14.0

The "filter-gain" property

```
  "filter-gain"               AgsPort~*
```

The filter gain port.

Owner: AgsFxHighPassChannel

Flags: Read / Write

Since: 3.13.0

The "q-lin" property

```
  "q-lin"                     AgsPort~*
```

The q-lin port.

Owner: AgsFxHighPassChannel

Flags: Read / Write

Since: 3.13.0

22.17 AgsFxHighPassChannelProcessor

AgsFxHighPassChannelProcessor — fx high pass channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxHighPassChannelProcessor *	ags_fx_high_pass_channel_processor_new ()
#define	AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR()
#define	AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_high_pass_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR
struct	AgsFxHighPassChannelProcessor
struct	AgsFxHighPassChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxHighPassChannelProcessor
```

Implemented Interfaces

AgsFxHighPassChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_high_pass_channel_processor.h>
```

Description

The AgsFxHighPassChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_high_pass_channel_processor_new ()

```
AgsFxHighPassChannelProcessor~*
ags_fx_high_pass_channel_processor_new
                                (AgsChannel *channel);
```

Create a new instance of AgsFxHighPassChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxHighPassChannelProcessor

Since: 3.13.0

AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR()

```
#define AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST ↩
    ((obj), AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR, AgsFxHighPassChannelProcessor))
```

AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR, AgsFxHighPassChannelProcessorClass))
```

AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS  ↩
    ((obj), AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR, AgsFxHighPassChannelProcessorClass))
```

AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR))
```

AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR))
```

ags_fx_high_pass_channel_processor_get_type ()

```
GType
ags_fx_high_pass_channel_processor_get_type
                                    ();
```

Types and Values

AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR                ( ↩
    ags_fx_high_pass_channel_processor_get_type())
```

struct AgsFxHighPassChannelProcessor

```
struct AgsFxHighPassChannelProcessor;
```

struct AgsFxHighPassChannelProcessorClass

```
struct AgsFxHighPassChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

22.18 AgsFxHighPassRecycling

AgsFxHighPassRecycling — fx high pass recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxHighPassRecycling *	ags_fx_high_pass_recycling_new ()
#define	AGS_FX_HIGH_PASS_RECYCLING()
#define	AGS_FX_HIGH_PASS_RECYCLING_CLASS()
#define	AGS_FX_HIGH_PASS_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_HIGH_PASS_RECYCLING()
#define	AGS_IS_FX_HIGH_PASS_RECYCLING_CLASS()
GType	ags_fx_high_pass_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_HIGH_PASS_RECYCLING
struct	AgsFxHighPassRecycling
struct	AgsFxHighPassRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxHighPassRecycling
```

Implemented Interfaces

AgsFxHighPassRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_high_pass_recycling.h>
```

Description

The AgsFxHighPassRecycling class provides ports to the effect processor.

Functions

ags_fx_high_pass_recycling_new ()

```
AgsFxHighPassRecycling~*
ags_fx_high_pass_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxHighPassRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxHighPassRecycling

Since: 3.13.0

AGS_FX_HIGH_PASS_RECYCLING()

```
#define AGS_FX_HIGH_PASS_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_RECYCLING, AgsFxHighPassRecycling))
```

AGS_FX_HIGH_PASS_RECYCLING_CLASS()

```
#define AGS_FX_HIGH_PASS_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_HIGH_PASS_RECYCLING, AgsFxHighPassRecyclingClass))
```

AGS_FX_HIGH_PASS_RECYCLING_GET_CLASS()

```
#define AGS_FX_HIGH_PASS_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_RECYCLING, AgsFxHighPassRecyclingClass))
```

AGS_IS_FX_HIGH_PASS_RECYCLING()

```
#define AGS_IS_FX_HIGH_PASS_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_HIGH_PASS_RECYCLING))
```

AGS_IS_FX_HIGH_PASS_RECYCLING_CLASS()

```
#define AGS_IS_FX_HIGH_PASS_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_HIGH_PASS_RECYCLING))
```

ags_fx_high_pass_recycling_get_type ()

```
GType
ags_fx_high_pass_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_HIGH_PASS_RECYCLING

```
#define AGS_TYPE_FX_HIGH_PASS_RECYCLING                (ags_fx_high_pass_recycling_get_type ↩
    ())
```

struct AgsFxHighPassRecycling

```
struct AgsFxHighPassRecycling;
```

struct AgsFxHighPassRecyclingClass

```
struct AgsFxHighPassRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

22.19 AgsFxEq10Audio

AgsFxEq10Audio — fx eq10 audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEq10Audio *	ags_fx_eq10_audio_new ()
#define	AGS_FX_EQ10_AUDIO()
#define	AGS_FX_EQ10_AUDIO_CLASS()
#define	AGS_FX_EQ10_AUDIO_GET_CLASS()
#define	AGS_IS_FX_EQ10_AUDIO()
#define	AGS_IS_FX_EQ10_AUDIO_CLASS()
GType	ags_fx_eq10_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_EQ10_AUDIO
struct	AgsFxEq10Audio
struct	AgsFxEq10AudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxEq10Audio
```

Implemented Interfaces

AgsFxEq10Audio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_eq10_audio.h>
```

Description

The AgsFxEq10Audio class provides ports to the effect processor.

Functions

ags_fx_eq10_audio_new ()

```
AgsFxEq10Audio~*
ags_fx_eq10_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxEq10Audio

Parameters

audio	the AgsAudio

Returns

the new AgsFxEq10Audio

Since: 3.3.0

AGS_FX_EQ10_AUDIO()

```
#define AGS_FX_EQ10_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO, AgsFxEq10Audio))
```

AGS_FX_EQ10_AUDIO_CLASS()

```
#define AGS_FX_EQ10_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_EQ10_AUDIO, AgsFxEq10AudioClass))
```

AGS_FX_EQ10_AUDIO_GET_CLASS()

```
#define AGS_FX_EQ10_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO, AgsFxEq10AudioClass))
```

AGS_IS_FX_EQ10_AUDIO()

```
#define AGS_IS_FX_EQ10_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO))
```

AGS_IS_FX_EQ10_AUDIO_CLASS()

```
#define AGS_IS_FX_EQ10_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_EQ10_AUDIO))
```

ags_fx_eq10_audio_get_type ()

```
GType
ags_fx_eq10_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_EQ10_AUDIO

```
#define AGS_TYPE_FX_EQ10_AUDIO                 (ags_fx_eq10_audio_get_type())
```

struct AgsFxEq10Audio

```
struct AgsFxEq10Audio;
```

struct AgsFxEq10AudioClass

```
struct AgsFxEq10AudioClass {
  AgsRecallAudioClass recall_audio;
};
```

22.20 AgsFxEq10AudioProcessor

AgsFxEq10AudioProcessor — fx eq10 audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEq10AudioProcessor *	ags_fx_eq10_audio_processor_new ()
#define	AGS_FX_EQ10_AUDIO_PROCESSOR()
#define	AGS_FX_EQ10_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_EQ10_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_EQ10_AUDIO_PROCESSOR()
#define	AGS_IS_FX_EQ10_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_eq10_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR
struct	AgsFxEq10AudioProcessor
struct	AgsFxEq10AudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxEq10AudioProcessor
```

Implemented Interfaces

AgsFxEq10AudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_eq10_audio_processor.h>
```

Description

The AgsFxEq10AudioProcessor class provides ports to the effect processor.

Functions

ags_fx_eq10_audio_processor_new ()

```
AgsFxEq10AudioProcessor~*
ags_fx_eq10_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxEq10AudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxEq10AudioProcessor

Since: 3.3.0

AGS_FX_EQ10_AUDIO_PROCESSOR()

```
#define AGS_FX_EQ10_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR, AgsFxEq10AudioProcessor))
```

AGS_FX_EQ10_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_EQ10_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR, AgsFxEq10AudioProcessorClass))
```

AGS_FX_EQ10_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_EQ10_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR, AgsFxEq10AudioProcessorClass))
```

AGS_IS_FX_EQ10_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_EQ10_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
     AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR))
```

AGS_IS_FX_EQ10_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_EQ10_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR))
```

ags_fx_eq10_audio_processor_get_type ()

```
GType
ags_fx_eq10_audio_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR                ( ↩
    ags_fx_eq10_audio_processor_get_type())
```

struct AgsFxEq10AudioProcessor

```
struct AgsFxEq10AudioProcessor;
```

struct AgsFxEq10AudioProcessorClass

```
struct AgsFxEq10AudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

22.21 AgsFxEq10AudioSignal

AgsFxEq10AudioSignal — fx eq10 audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEq10AudioSignal *	ags_fx_eq10_audio_signal_new ()
#define	AGS_FX_EQ10_AUDIO_SIGNAL()
#define	AGS_FX_EQ10_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_EQ10_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_EQ10_AUDIO_SIGNAL()
#define	AGS_IS_FX_EQ10_AUDIO_SIGNAL_CLASS()
GType	ags_fx_eq10_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_EQ10_AUDIO_SIGNAL
struct	AgsFxEq10AudioSignal
struct	AgsFxEq10AudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxEq10AudioSignal
```

Implemented Interfaces

AgsFxEq10AudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_eq10_audio_signal.h>
```

Description

The AgsFxEq10AudioSignal class provides ports to the effect processor.

Functions

ags_fx_eq10_audio_signal_new ()

```
AgsFxEq10AudioSignal~*
ags_fx_eq10_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxEq10AudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxEq10AudioSignal

Since: 3.3.0

AGS_FX_EQ10_AUDIO_SIGNAL()

```
#define AGS_FX_EQ10_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_SIGNAL, AgsFxEq10AudioSignal))
```

AGS_FX_EQ10_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_EQ10_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_SIGNAL, AgsFxEq10AudioSignalClass))
```

AGS_FX_EQ10_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_EQ10_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_SIGNAL, AgsFxEq10AudioSignalClass))
```

AGS_IS_FX_EQ10_AUDIO_SIGNAL()

```
#define AGS_IS_FX_EQ10_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_SIGNAL))
```

AGS_IS_FX_EQ10_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_EQ10_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_EQ10_AUDIO_SIGNAL))
```

ags_fx_eq10_audio_signal_get_type ()

```
GType
ags_fx_eq10_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_EQ10_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_EQ10_AUDIO_SIGNAL                (ags_fx_eq10_audio_signal_get_type())
```

struct AgsFxEq10AudioSignal

```
struct AgsFxEq10AudioSignal;
```

struct AgsFxEq10AudioSignalClass

```
struct AgsFxEq10AudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

22.22 AgsFxEq10Channel

AgsFxEq10Channel — fx eq10 channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_EQ10_CHANNEL_INPUT_DATA()
#define	AGS_FX_EQ10_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxEq10ChannelInputData *	ags_fx_eq10_channel_input_data_alloc ()
void	ags_fx_eq10_channel_input_data_free ()
AgsFxEq10Channel *	ags_fx_eq10_channel_new ()
#define	AGS_FX_EQ10_CHANNEL()
#define	AGS_FX_EQ10_CHANNEL_CLASS()
#define	AGS_FX_EQ10_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_EQ10_CHANNEL()
#define	AGS_IS_FX_EQ10_CHANNEL_CLASS()
GType	ags_fx_eq10_channel_get_type ()

Properties

AgsPort *	peak-112hz	Read / Write
AgsPort *	peak-14336hz	Read / Write
AgsPort *	peak-1792hz	Read / Write
AgsPort *	peak-224hz	Read / Write
AgsPort *	peak-28hz	Read / Write
AgsPort *	peak-3584hz	Read / Write
AgsPort *	peak-448hz	Read / Write
AgsPort *	peak-56hz	Read / Write

AgsPort *	peak-7168hz	Read / Write
AgsPort *	peak-896hz	Read / Write
AgsPort *	pressure	Read / Write

Types and Values

#define	AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE
struct	AgsFxEq10ChannelInputData
#define	AGS_TYPE_FX_EQ10_CHANNEL
struct	AgsFxEq10Channel
struct	AgsFxEq10ChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxEq10Channel
```

Implemented Interfaces

AgsFxEq10Channel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_eq10_channel.h>
```

Description

The AgsFxEq10Channel class provides ports to the effect processor.

Functions

AGS_FX_EQ10_CHANNEL_INPUT_DATA()

```
#define AGS_FX_EQ10_CHANNEL_INPUT_DATA(ptr) ((AgsFxEq10ChannelInputData *)(ptr))
```

AGS_FX_EQ10_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_EQ10_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&(((AgsFxEq10ChannelInputData ↩
    *)(ptr))->strct_mutex))
```

ags_fx_eq10_channel_input_data_alloc ()

```
AgsFxEq10ChannelInputData~*
ags_fx_eq10_channel_input_data_alloc ();
```

Allocate AgsFxEq10ChannelInputData

Returns

the new AgsFxEq10ChannelInputData.

[type gpointer][transfer full]

Since: 3.3.0

ags_fx_eq10_channel_input_data_free ()

```
void
ags_fx_eq10_channel_input_data_free (AgsFxEq10ChannelInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxEq10ChannelInputData.	*[type gpointer][transfer full]*

Since: 3.3.0

ags_fx_eq10_channel_new ()

```
AgsFxEq10Channel~*
ags_fx_eq10_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxEq10Channel

Parameters

channel	the AgsChannel	

Returns

the new AgsFxEq10Channel

Since: 3.3.0

AGS_FX_EQ10_CHANNEL()

```
#define AGS_FX_EQ10_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_EQ10_CHANNEL, AgsFxEq10Channel))
```

AGS_FX_EQ10_CHANNEL_CLASS()

```
#define AGS_FX_EQ10_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_EQ10_CHANNEL, AgsFxEq10ChannelClass))
```

AGS_FX_EQ10_CHANNEL_GET_CLASS()

```
#define AGS_FX_EQ10_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_EQ10_CHANNEL, AgsFxEq10ChannelClass))
```

AGS_IS_FX_EQ10_CHANNEL()

```
#define AGS_IS_FX_EQ10_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_EQ10_CHANNEL))
```

AGS_IS_FX_EQ10_CHANNEL_CLASS()

```
#define AGS_IS_FX_EQ10_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_EQ10_CHANNEL))
```

ags_fx_eq10_channel_get_type ()

```
GType
ags_fx_eq10_channel_get_type ();
```

Types and Values

AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE

```
#define AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE (8)
```

struct AgsFxEq10ChannelInputData

```
struct AgsFxEq10ChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  gdouble cache_28hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_56hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_112hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_224hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_448hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_896hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_1792hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_3584hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_7168hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];
  gdouble cache_14336hz[AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE];

  gdouble *output;
  gdouble *input;
};
```

AGS_TYPE_FX_EQ10_CHANNEL

```
#define AGS_TYPE_FX_EQ10_CHANNEL                (ags_fx_eq10_channel_get_type())
```

struct AgsFxEq10Channel

```
struct AgsFxEq10Channel;
```

struct AgsFxEq10ChannelClass

```
struct AgsFxEq10ChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "peak-112hz" property

```
  "peak-112hz"                AgsPort~*
```

The peak 112Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-14336hz" property

```
  "peak-14336hz"              AgsPort~*
```

The peak 14336Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-1792hz" property

```
  "peak-1792hz"               AgsPort~*
```

The peak 1792Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-224hz" property

```
  "peak-224hz"                AgsPort~*
```

The peak 224Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-28hz" property

```
"peak-28hz"                 AgsPort~*
```

The peak 28Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-3584hz" property

```
"peak-3584hz"               AgsPort~*
```

The peak 3584Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-448hz" property

```
"peak-448hz"                AgsPort~*
```

The peak 448Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-56hz" property

```
"peak-56hz"                 AgsPort~*
```

The peak 56Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-7168hz" property

```
"peak-7168hz"               AgsPort~*
```

The peak 7168Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "peak-896hz" property

```
"peak-896hz"                AgsPort~*
```

The peak 896Hz port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

The "pressure" property

```
"pressure"                  AgsPort~*
```

The pressure port.

Owner: AgsFxEq10Channel

Flags: Read / Write

Since: 3.3.0

22.23 AgsFxEq10ChannelProcessor

AgsFxEq10ChannelProcessor — fx eq10 channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEq10ChannelProcessor *	ags_fx_eq10_channel_processor_new ()
#define	AGS_FX_EQ10_CHANNEL_PROCESSOR()
#define	AGS_FX_EQ10_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_EQ10_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_EQ10_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_EQ10_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_eq10_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR
struct	AgsFxEq10ChannelProcessor
struct	AgsFxEq10ChannelProcessorClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
        &#x2570;&#x2500;&#x2500; AgsFxEq10ChannelProcessor
```

Implemented Interfaces

AgsFxEq10ChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_eq10_channel_processor.h>
```

Description

The AgsFxEq10ChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_eq10_channel_processor_new ()

```
AgsFxEq10ChannelProcessor~*
ags_fx_eq10_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxEq10ChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxEq10ChannelProcessor

Since: 3.3.0

AGS_FX_EQ10_CHANNEL_PROCESSOR()

```
#define AGS_FX_EQ10_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR, AgsFxEq10ChannelProcessor))
```

AGS_FX_EQ10_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_EQ10_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR, AgsFxEq10ChannelProcessorClass))
```

AGS_FX_EQ10_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_EQ10_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR, AgsFxEq10ChannelProcessorClass))
```

AGS_IS_FX_EQ10_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_EQ10_CHANNEL_PROCESSOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR))
```

AGS_IS_FX_EQ10_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_EQ10_CHANNEL_PROCESSOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR))
```

ags_fx_eq10_channel_processor_get_type ()

```
GType
ags_fx_eq10_channel_processor_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR                 ( ↩
    ags_fx_eq10_channel_processor_get_type())
```

struct AgsFxEq10ChannelProcessor

```
struct AgsFxEq10ChannelProcessor;
```

struct AgsFxEq10ChannelProcessorClass

```
struct AgsFxEq10ChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

22.24 AgsFxEq10Recycling

AgsFxEq10Recycling — fx eq10 recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEq10Recycling *	ags_fx_eq10_recycling_new ()
#define	AGS_FX_EQ10_RECYCLING()
#define	AGS_FX_EQ10_RECYCLING_CLASS()
#define	AGS_FX_EQ10_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_EQ10_RECYCLING()
#define	AGS_IS_FX_EQ10_RECYCLING_CLASS()
GType	ags_fx_eq10_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_EQ10_RECYCLING
struct	AgsFxEq10Recycling
struct	AgsFxEq10RecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxEq10Recycling
```

Implemented Interfaces

AgsFxEq10Recycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_eq10_recycling.h>
```

Description

The AgsFxEq10Recycling class provides ports to the effect processor.

Functions

ags_fx_eq10_recycling_new ()

```
AgsFxEq10Recycling~*
ags_fx_eq10_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxEq10Recycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxEq10Recycling

Since: 3.3.0

AGS_FX_EQ10_RECYCLING()

```
#define AGS_FX_EQ10_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_EQ10_RECYCLING, AgsFxEq10Recycling))
```

AGS_FX_EQ10_RECYCLING_CLASS()

```
#define AGS_FX_EQ10_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_EQ10_RECYCLING, AgsFxEq10RecyclingClass))
```

AGS_FX_EQ10_RECYCLING_GET_CLASS()

```
#define AGS_FX_EQ10_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_EQ10_RECYCLING, AgsFxEq10RecyclingClass))
```

AGS_IS_FX_EQ10_RECYCLING()

```
#define AGS_IS_FX_EQ10_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_EQ10_RECYCLING))
```

AGS_IS_FX_EQ10_RECYCLING_CLASS()

```
#define AGS_IS_FX_EQ10_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_EQ10_RECYCLING))
```

ags_fx_eq10_recycling_get_type ()

```
GType
ags_fx_eq10_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_EQ10_RECYCLING

```
#define AGS_TYPE_FX_EQ10_RECYCLING                (ags_fx_eq10_recycling_get_type())
```

struct AgsFxEq10Recycling

```
struct AgsFxEq10Recycling;
```

struct AgsFxEq10RecyclingClass

```
struct AgsFxEq10RecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

22.25 AgsFxEnvelopeAudio

AgsFxEnvelopeAudio — fx envelope audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEnvelopeAudio *	ags_fx_envelope_audio_new ()
#define	AGS_FX_ENVELOPE_AUDIO()
#define	AGS_FX_ENVELOPE_AUDIO_CLASS()
#define	AGS_FX_ENVELOPE_AUDIO_GET_CLASS()
#define	AGS_IS_FX_ENVELOPE_AUDIO()
#define	AGS_IS_FX_ENVELOPE_AUDIO_CLASS()
GType	ags_fx_envelope_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_ENVELOPE_AUDIO
struct	AgsFxEnvelopeAudio
struct	AgsFxEnvelopeAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxEnvelopeAudio
```

Implemented Interfaces

AgsFxEnvelopeAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_envelope_audio.h>
```

Description

The AgsFxEnvelopeAudio class provides ports to the effect processor.

Functions

ags_fx_envelope_audio_new ()

```
AgsFxEnvelopeAudio~*
ags_fx_envelope_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxEnvelopeAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxEnvelopeAudio

Since: 3.3.0

AGS_FX_ENVELOPE_AUDIO()

```
#define AGS_FX_ENVELOPE_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_ENVELOPE_AUDIO, AgsFxEnvelopeAudio))
```

AGS_FX_ENVELOPE_AUDIO_CLASS()

```
#define AGS_FX_ENVELOPE_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_ENVELOPE_AUDIO, AgsFxEnvelopeAudioClass))
```

AGS_FX_ENVELOPE_AUDIO_GET_CLASS()

```
#define AGS_FX_ENVELOPE_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_ENVELOPE_AUDIO, AgsFxEnvelopeAudioClass))
```

AGS_IS_FX_ENVELOPE_AUDIO()

```
#define AGS_IS_FX_ENVELOPE_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_ENVELOPE_AUDIO))
```

AGS_IS_FX_ENVELOPE_AUDIO_CLASS()

```
#define AGS_IS_FX_ENVELOPE_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_ENVELOPE_AUDIO))
```

ags_fx_envelope_audio_get_type ()

```
GType
ags_fx_envelope_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_ENVELOPE_AUDIO

```
#define AGS_TYPE_FX_ENVELOPE_AUDIO                (ags_fx_envelope_audio_get_type())
```

struct AgsFxEnvelopeAudio

```
struct AgsFxEnvelopeAudio;
```

struct AgsFxEnvelopeAudioClass

```
struct AgsFxEnvelopeAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

22.26 AgsFxEnvelopeAudioProcessor

AgsFxEnvelopeAudioProcessor — fx envelope audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEnvelopeAudioProcessor *	ags_fx_envelope_audio_processor_new ()
#define	AGS_FX_ENVELOPE_AUDIO_PROCESSOR()
#define	AGS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_ENVELOPE_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR()
#define	AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_envelope_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR
struct	AgsFxEnvelopeAudioProcessor
struct	AgsFxEnvelopeAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxEnvelopeAudioProcessor
```

Implemented Interfaces

AgsFxEnvelopeAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_envelope_audio_processor.h>
```

Description

The AgsFxEnvelopeAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_envelope_audio_processor_new ()

```
AgsFxEnvelopeAudioProcessor~*
ags_fx_envelope_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxEnvelopeAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxEnvelopeAudioProcessor

Since: 3.3.0

AGS_FX_ENVELOPE_AUDIO_PROCESSOR()

```
#define AGS_FX_ENVELOPE_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR, AgsFxEnvelopeAudioProcessor))
```

AGS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR, AgsFxEnvelopeAudioProcessorClass))
```

AGS_FX_ENVELOPE_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_ENVELOPE_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR, AgsFxEnvelopeAudioProcessorClass))
```

AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR))
```

AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR))
```

ags_fx_envelope_audio_processor_get_type ()

```
GType
ags_fx_envelope_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR                ( ↩
    ags_fx_envelope_audio_processor_get_type())
```

struct AgsFxEnvelopeAudioProcessor

```
struct AgsFxEnvelopeAudioProcessor;
```

struct AgsFxEnvelopeAudioProcessorClass

```
struct AgsFxEnvelopeAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

22.27 AgsFxEnvelopeAudioSignal

AgsFxEnvelopeAudioSignal — fx envelope audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEnvelopeAudioSignal *	ags_fx_envelope_audio_signal_new ()
#define	AGS_FX_ENVELOPE_AUDIO_SIGNAL()
#define	AGS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_ENVELOPE_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL()
#define	AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS()
GType	ags_fx_envelope_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL
struct	AgsFxEnvelopeAudioSignal
struct	AgsFxEnvelopeAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxEnvelopeAudioSignal
```

Implemented Interfaces

AgsFxEnvelopeAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_envelope_audio_signal.h>
```

Description

The AgsFxEnvelopeAudioSignal class provides ports to the effect processor.

Functions

ags_fx_envelope_audio_signal_new ()

```
AgsFxEnvelopeAudioSignal~*
ags_fx_envelope_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxEnvelopeAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxEnvelopeAudioSignal

Since: 3.3.0

AGS_FX_ENVELOPE_AUDIO_SIGNAL()

```
#define AGS_FX_ENVELOPE_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL, AgsFxEnvelopeAudioSignal))
```

AGS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL, AgsFxEnvelopeAudioSignalClass))
```

AGS_FX_ENVELOPE_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_ENVELOPE_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL, AgsFxEnvelopeAudioSignalClass))
```

AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL()

```
#define AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL))
```

AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL))
```

ags_fx_envelope_audio_signal_get_type ()

```
GType
ags_fx_envelope_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL                ( ↩
    ags_fx_envelope_audio_signal_get_type())
```

struct AgsFxEnvelopeAudioSignal

```
struct AgsFxEnvelopeAudioSignal;
```

struct AgsFxEnvelopeAudioSignalClass

```
struct AgsFxEnvelopeAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

22.28 AgsFxEnvelopeChannel

AgsFxEnvelopeChannel — fx envelope channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEnvelopeChannel *	ags_fx_envelope_channel_new ()
#define	AGS_FX_ENVELOPE_CHANNEL()
#define	AGS_FX_ENVELOPE_CHANNEL_CLASS()
#define	AGS_FX_ENVELOPE_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_ENVELOPE_CHANNEL()
#define	AGS_IS_FX_ENVELOPE_CHANNEL_CLASS()
GType	ags_fx_envelope_channel_get_type ()

Properties

AgsPort *	attack	Read / Write
AgsPort *	decay	Read / Write
AgsPort *	fixed-length	Read / Write
AgsPort *	ratio	Read / Write
AgsPort *	release	Read / Write
AgsPort *	sustain	Read / Write

Types and Values

enum	AgsFxEnvelopeChannelMode
#define	AGS_TYPE_FX_ENVELOPE_CHANNEL
struct	AgsFxEnvelopeChannel
struct	AgsFxEnvelopeChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxEnvelopeChannel
```

Implemented Interfaces

AgsFxEnvelopeChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_envelope_channel.h>
```

Description

The AgsFxEnvelopeChannel class provides ports to the effect processor.

Functions

ags_fx_envelope_channel_new ()

```
AgsFxEnvelopeChannel~*
ags_fx_envelope_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxEnvelopeChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxEnvelopeChannel

Since: 3.3.0

AGS_FX_ENVELOPE_CHANNEL()

```
#define AGS_FX_ENVELOPE_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_ENVELOPE_CHANNEL, AgsFxEnvelopeChannel))
```

AGS_FX_ENVELOPE_CHANNEL_CLASS()

```
#define AGS_FX_ENVELOPE_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_ENVELOPE_CHANNEL, AgsFxEnvelopeChannelClass))
```

AGS_FX_ENVELOPE_CHANNEL_GET_CLASS()

```
#define AGS_FX_ENVELOPE_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_ENVELOPE_CHANNEL, AgsFxEnvelopeChannelClass))
```

AGS_IS_FX_ENVELOPE_CHANNEL()

```
#define AGS_IS_FX_ENVELOPE_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_ENVELOPE_CHANNEL))
```

AGS_IS_FX_ENVELOPE_CHANNEL_CLASS()

```
#define AGS_IS_FX_ENVELOPE_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_ENVELOPE_CHANNEL))
```

ags_fx_envelope_channel_get_type ()

```
GType
ags_fx_envelope_channel_get_type ();
```

Types and Values

enum AgsFxEnvelopeChannelMode

Members

AGS_FX_ENVELOPE_CHANNEL_MODE_STREAM_LENGTH		
AGS_FX_ENVELOPE_CHANNEL_MODE_NOTE_LENGTH		
AGS_FX_ENVELOPE_CHANNEL_MODE_FIXED_LENGTH		

AGS_TYPE_FX_ENVELOPE_CHANNEL

```
#define AGS_TYPE_FX_ENVELOPE_CHANNEL                (ags_fx_envelope_channel_get_type())
```

struct AgsFxEnvelopeChannel

```
struct AgsFxEnvelopeChannel;
```

struct AgsFxEnvelopeChannelClass

```
struct AgsFxEnvelopeChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "attack" property

```
  "attack"                    AgsPort~*
```

The beats per minute.

Owner: AgsFxEnvelopeChannel

Flags: Read / Write

Since: 3.3.0

The "decay" property

```
"decay"                    AgsPort~*
```

The beats per minute.

Owner: AgsFxEnvelopeChannel

Flags: Read / Write

Since: 3.3.0

The "fixed-length" property

```
"fixed-length"             AgsPort~*
```

The beats per minute.

Owner: AgsFxEnvelopeChannel

Flags: Read / Write

Since: 3.3.0

The "ratio" property

```
"ratio"                    AgsPort~*
```

The beats per minute.

Owner: AgsFxEnvelopeChannel

Flags: Read / Write

Since: 3.3.0

The "release" property

```
"release"                  AgsPort~*
```

The beats per minute.

Owner: AgsFxEnvelopeChannel

Flags: Read / Write

Since: 3.3.0

The "sustain" property

```
"sustain"                  AgsPort~*
```

The beats per minute.

Owner: AgsFxEnvelopeChannel

Flags: Read / Write

Since: 3.3.0

22.29 AgsFxEnvelopeChannelProcessor

AgsFxEnvelopeChannelProcessor — fx envelope channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEnvelopeChannelProcessor *	ags_fx_envelope_channel_processor_new ()
#define	AGS_FX_ENVELOPE_CHANNEL_PROCESSOR()
#define	AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_envelope_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR
struct	AgsFxEnvelopeChannelProcessor
struct	AgsFxEnvelopeChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxEnvelopeChannelProcessor
```

Implemented Interfaces

AgsFxEnvelopeChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_envelope_channel_processor.h>
```

Description

The AgsFxEnvelopeChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_envelope_channel_processor_new ()

```
AgsFxEnvelopeChannelProcessor~*
ags_fx_envelope_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxEnvelopeChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxEnvelopeChannelProcessor

Since: 3.3.0

AGS_FX_ENVELOPE_CHANNEL_PROCESSOR()

```
#define AGS_FX_ENVELOPE_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR, AgsFxEnvelopeChannelProcessor))
```

AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR, AgsFxEnvelopeChannelProcessorClass))
```

AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR, AgsFxEnvelopeChannelProcessorClass))
```

AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE  ↩
    ((obj), AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR))
```

AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR))
```

ags_fx_envelope_channel_processor_get_type ()

```
GType
ags_fx_envelope_channel_processor_get_type
                                  ();
```

Types and Values

AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR                ( ↩
    ags_fx_envelope_channel_processor_get_type())
```

struct AgsFxEnvelopeChannelProcessor

```
struct AgsFxEnvelopeChannelProcessor;
```

struct AgsFxEnvelopeChannelProcessorClass

```
struct AgsFxEnvelopeChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

22.30 AgsFxEnvelopeRecycling

AgsFxEnvelopeRecycling — fx envelope recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxEnvelopeRecycling *	ags_fx_envelope_recycling_new ()
#define	AGS_FX_ENVELOPE_RECYCLING()
#define	AGS_FX_ENVELOPE_RECYCLING_CLASS()
#define	AGS_FX_ENVELOPE_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_ENVELOPE_RECYCLING()
#define	AGS_IS_FX_ENVELOPE_RECYCLING_CLASS()
GType	ags_fx_envelope_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_ENVELOPE_RECYCLING
struct	AgsFxEnvelopeRecycling
struct	AgsFxEnvelopeRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxEnvelopeRecycling
```

Implemented Interfaces

AgsFxEnvelopeRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_envelope_recycling.h>
```

Description

The AgsFxEnvelopeRecycling class provides ports to the effect processor.

Functions

ags_fx_envelope_recycling_new ()

```
AgsFxEnvelopeRecycling~*
ags_fx_envelope_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxEnvelopeRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxEnvelopeRecycling

Since: 3.3.0

AGS_FX_ENVELOPE_RECYCLING()

```
#define AGS_FX_ENVELOPE_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_FX_ENVELOPE_RECYCLING, AgsFxEnvelopeRecycling))
```

AGS_FX_ENVELOPE_RECYCLING_CLASS()

```
#define AGS_FX_ENVELOPE_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
    AGS_TYPE_FX_ENVELOPE_RECYCLING, AgsFxEnvelopeRecyclingClass))
```

AGS_FX_ENVELOPE_RECYCLING_GET_CLASS()

```
#define AGS_FX_ENVELOPE_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
    AGS_TYPE_FX_ENVELOPE_RECYCLING, AgsFxEnvelopeRecyclingClass))
```

AGS_IS_FX_ENVELOPE_RECYCLING()

```
#define AGS_IS_FX_ENVELOPE_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), ↩
    AGS_TYPE_FX_ENVELOPE_RECYCLING))
```

AGS_IS_FX_ENVELOPE_RECYCLING_CLASS()

```
#define AGS_IS_FX_ENVELOPE_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
    AGS_TYPE_FX_ENVELOPE_RECYCLING))
```

ags_fx_envelope_recycling_get_type ()

```
GType
ags_fx_envelope_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_ENVELOPE_RECYCLING

```
#define AGS_TYPE_FX_ENVELOPE_RECYCLING                (ags_fx_envelope_recycling_get_type() ↩
    )
```

struct AgsFxEnvelopeRecycling

```
struct AgsFxEnvelopeRecycling;
```

struct AgsFxEnvelopeRecyclingClass

```
struct AgsFxEnvelopeRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

Chapter 23

Fx Engine - misc plugins

Miscellaneous plugins to process audio data.

23.1 AgsFxLadspaAudio

AgsFxLadspaAudio — fx ladspa audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLadspaAudio *	ags_fx_ladspa_audio_new ()
#define	AGS_FX_LADSPA_AUDIO()
#define	AGS_FX_LADSPA_AUDIO_CLASS()
#define	AGS_FX_LADSPA_AUDIO_GET_CLASS()
#define	AGS_IS_FX_LADSPA_AUDIO()
#define	AGS_IS_FX_LADSPA_AUDIO_CLASS()
GType	ags_fx_ladspa_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_LADSPA_AUDIO
struct	AgsFxLadspaAudio
struct	AgsFxLadspaAudioClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallAudio
        &#x2570;&#x2500;&#x2500; AgsFxLadspaAudio
```

Implemented Interfaces

AgsFxLadspaAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_ladspa_audio.h>
```

Description

The AgsFxLadspaAudio class provides ports to the effect processor.

Functions

ags_fx_ladspa_audio_new ()

```
AgsFxLadspaAudio~*
ags_fx_ladspa_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxLadspaAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxLadspaAudio

Since: 3.3.0

AGS_FX_LADSPA_AUDIO()

```
#define AGS_FX_LADSPA_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO, AgsFxLadspaAudio))
```

AGS_FX_LADSPA_AUDIO_CLASS()

```
#define AGS_FX_LADSPA_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO, AgsFxLadspaAudioClass))
```

AGS_FX_LADSPA_AUDIO_GET_CLASS()

```
#define AGS_FX_LADSPA_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO, AgsFxLadspaAudioClass))
```

AGS_IS_FX_LADSPA_AUDIO()

```
#define AGS_IS_FX_LADSPA_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO))
```

AGS_IS_FX_LADSPA_AUDIO_CLASS()

```
#define AGS_IS_FX_LADSPA_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO))
```

ags_fx_ladspa_audio_get_type ()

```
GType
ags_fx_ladspa_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_LADSPA_AUDIO

```
#define AGS_TYPE_FX_LADSPA_AUDIO                 (ags_fx_ladspa_audio_get_type())
```

struct AgsFxLadspaAudio

```
struct AgsFxLadspaAudio;
```

struct AgsFxLadspaAudioClass

```
struct AgsFxLadspaAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

23.2 AgsFxLadspaAudioProcessor

AgsFxLadspaAudioProcessor — fx ladspa audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLadspaAudioProcessor *	ags_fx_ladspa_audio_processor_new ()
#define	AGS_FX_LADSPA_AUDIO_PROCESSOR()
#define	AGS_FX_LADSPA_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_LADSPA_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LADSPA_AUDIO_PROCESSOR()
#define	AGS_IS_FX_LADSPA_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_ladspa_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR
struct	AgsFxLadspaAudioProcessor
struct	AgsFxLadspaAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxLadspaAudioProcessor
```

Implemented Interfaces

AgsFxLadspaAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_ladspa_audio_processor.h>
```

Description

The AgsFxLadspaAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_ladspa_audio_processor_new ()

```
AgsFxLadspaAudioProcessor~*
ags_fx_ladspa_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxLadspaAudioProcessor

Parameters

audio	the AgsAudio	

Returns

the new AgsFxLadspaAudioProcessor

Since: 3.3.0

AGS_FX_LADSPA_AUDIO_PROCESSOR()

```
#define AGS_FX_LADSPA_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj) ↩
    , AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR, AgsFxLadspaAudioProcessor))
```

AGS_FX_LADSPA_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_LADSPA_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), ↩
     AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR, AgsFxLadspaAudioProcessorClass))
```

AGS_FX_LADSPA_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LADSPA_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj) ↩
    , AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR, AgsFxLadspaAudioProcessorClass))
```

AGS_IS_FX_LADSPA_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_LADSPA_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj ↩
    ), AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR))
```

AGS_IS_FX_LADSPA_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LADSPA_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class) ↩
    , AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR))
```

ags_fx_ladspa_audio_processor_get_type ()

```
GType
ags_fx_ladspa_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR                ( ↩
    ags_fx_ladspa_audio_processor_get_type())
```

struct AgsFxLadspaAudioProcessor

```
struct AgsFxLadspaAudioProcessor;
```

struct AgsFxLadspaAudioProcessorClass

```
struct AgsFxLadspaAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

23.3 AgsFxLadspaAudioSignal

AgsFxLadspaAudioSignal — fx ladspa audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLadspaAudioSignal *	ags_fx_ladspa_audio_signal_new ()
#define	AGS_FX_LADSPA_AUDIO_SIGNAL()
#define	AGS_FX_LADSPA_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_LADSPA_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_LADSPA_AUDIO_SIGNAL()
#define	AGS_IS_FX_LADSPA_AUDIO_SIGNAL_CLASS()
GType	ags_fx_ladspa_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL
struct	AgsFxLadspaAudioSignal
struct	AgsFxLadspaAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxLadspaAudioSignal
```

Implemented Interfaces

AgsFxLadspaAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_ladspa_audio_signal.h>
```

Description

The AgsFxLadspaAudioSignal class provides ports to the effect processor.

Functions

ags_fx_ladspa_audio_signal_new ()

```
AgsFxLadspaAudioSignal~*
ags_fx_ladspa_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxLadspaAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxLadspaAudioSignal

Since: 3.3.0

AGS_FX_LADSPA_AUDIO_SIGNAL()

```
#define AGS_FX_LADSPA_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL, AgsFxLadspaAudioSignal))
```

AGS_FX_LADSPA_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_LADSPA_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL, AgsFxLadspaAudioSignalClass))
```

AGS_FX_LADSPA_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_LADSPA_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL, AgsFxLadspaAudioSignalClass))
```

AGS_IS_FX_LADSPA_AUDIO_SIGNAL()

```
#define AGS_IS_FX_LADSPA_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL))
```

AGS_IS_FX_LADSPA_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_LADSPA_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL))
```

ags_fx_ladspa_audio_signal_get_type ()

```
GType
ags_fx_ladspa_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL                (ags_fx_ladspa_audio_signal_get_type ↩
    ())
```

struct AgsFxLadspaAudioSignal

```
struct AgsFxLadspaAudioSignal;
```

struct AgsFxLadspaAudioSignalClass

```
struct AgsFxLadspaAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

23.4 AgsFxLadspaChannel

AgsFxLadspaChannel — fx ladspa channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_LADSPA_CHANNEL_INPUT_DATA()
#define	AGS_FX_LADSPA_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxLadspaChannelInputData *	ags_fx_ladspa_channel_input_data_alloc ()
void	ags_fx_ladspa_channel_input_data_free ()
void	ags_fx_ladspa_channel_load_plugin ()
void	ags_fx_ladspa_channel_load_port ()
AgsFxLadspaChannel *	ags_fx_ladspa_channel_new ()
#define	AGS_FX_LADSPA_CHANNEL()
#define	AGS_FX_LADSPA_CHANNEL_CLASS()
#define	AGS_FX_LADSPA_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_LADSPA_CHANNEL()
#define	AGS_IS_FX_LADSPA_CHANNEL_CLASS()
GType	ags_fx_ladspa_channel_get_type ()

Types and Values

struct	AgsFxLadspaChannelInputData
#define	AGS_TYPE_FX_LADSPA_CHANNEL
struct	AgsFxLadspaChannel
struct	AgsFxLadspaChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxLadspaChannel
```

Implemented Interfaces

AgsFxLadspaChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_ladspa_channel.h>
```

Description

The AgsFxLadspaChannel class provides ports to the effect processor.

Functions

AGS_FX_LADSPA_CHANNEL_INPUT_DATA()

```
#define AGS_FX_LADSPA_CHANNEL_INPUT_DATA(ptr) ((AgsFxLadspaChannelInputData *) (ptr))
```

AGS_FX_LADSPA_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_LADSPA_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&((( ↩
    AgsFxLadspaChannelInputData *)(ptr))->strct_mutex))
```

ags_fx_ladspa_channel_input_data_alloc ()

```
AgsFxLadspaChannelInputData~*
ags_fx_ladspa_channel_input_data_alloc
                                ();
```

Allocate AgsFxLadspaChannelInputData

Returns

the new AgsFxLadspaChannelInputData

Since: 3.3.0

ags_fx_ladspa_channel_input_data_free ()

```
void
ags_fx_ladspa_channel_input_data_free (AgsFxLadspaChannelInputData *input_data);
```

Free *input_data*.

Parameters

input_data	the AgsFxLadspaChannelInputData

Since: 3.3.0

ags_fx_ladspa_channel_load_plugin ()

```
void
ags_fx_ladspa_channel_load_plugin (AgsFxLadspaChannel *fx_ladspa_channel);
```

Load plugin of *fx_ladspa_channel* .

Parameters

fx_ladspa_channel	the AgsFxLadspaChannel

Since: 3.3.0

ags_fx_ladspa_channel_load_port ()

```
void
ags_fx_ladspa_channel_load_port (AgsFxLadspaChannel *fx_ladspa_channel);
```

Load port of *fx_ladspa_channel* .

Parameters

fx_ladspa_channel	the AgsFxLadspaChannel

Since: 3.3.0

ags_fx_ladspa_channel_new ()

```
AgsFxLadspaChannel~*
ags_fx_ladspa_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxLadspaChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxLadspaChannel

Since: 3.3.0

AGS_FX_LADSPA_CHANNEL()

```
#define AGS_FX_LADSPA_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LADSPA_CHANNEL, AgsFxLadspaChannel))
```

AGS_FX_LADSPA_CHANNEL_CLASS()

```
#define AGS_FX_LADSPA_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LADSPA_CHANNEL, AgsFxLadspaChannelClass))
```

AGS_FX_LADSPA_CHANNEL_GET_CLASS()

```
#define AGS_FX_LADSPA_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LADSPA_CHANNEL, AgsFxLadspaChannelClass))
```

AGS_IS_FX_LADSPA_CHANNEL()

```
#define AGS_IS_FX_LADSPA_CHANNEL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LADSPA_CHANNEL))
```

AGS_IS_FX_LADSPA_CHANNEL_CLASS()

```
#define AGS_IS_FX_LADSPA_CHANNEL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LADSPA_CHANNEL))
```

ags_fx_ladspa_channel_get_type ()

```
GType
ags_fx_ladspa_channel_get_type ();
```

Types and Values

struct AgsFxLadspaChannelInputData

```
struct AgsFxLadspaChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  LADSPA_Data *output;
  LADSPA_Data *input;

  LADSPA_Handle ladspa_handle;
};
```

AGS_TYPE_FX_LADSPA_CHANNEL

```
#define AGS_TYPE_FX_LADSPA_CHANNEL                (ags_fx_ladspa_channel_get_type())
```

struct AgsFxLadspaChannel

```
struct AgsFxLadspaChannel;
```

struct AgsFxLadspaChannelClass

```
struct AgsFxLadspaChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

23.5 AgsFxLadspaChannelProcessor

AgsFxLadspaChannelProcessor — fx ladspa channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLadspaChannelProcessor *	ags_fx_ladspa_channel_processor_new ()
#define	AGS_FX_LADSPA_CHANNEL_PROCESSOR()
#define	AGS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_LADSPA_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_ladspa_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR
struct	AgsFxLadspaChannelProcessor
struct	AgsFxLadspaChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxLadspaChannelProcessor
```

Implemented Interfaces

AgsFxLadspaChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_ladspa_channel_processor.h>
```

Description

The AgsFxLadspaChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_ladspa_channel_processor_new ()

```
AgsFxLadspaChannelProcessor~*
ags_fx_ladspa_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxLadspaChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxLadspaChannelProcessor

Since: 3.3.0

AGS_FX_LADSPA_CHANNEL_PROCESSOR()

```
#define AGS_FX_LADSPA_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR, AgsFxLadspaChannelProcessor))
```

AGS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class ↩
    ), AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR, AgsFxLadspaChannelProcessorClass))
```

AGS_FX_LADSPA_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LADSPA_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR, AgsFxLadspaChannelProcessorClass))
```

AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR))
```

AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR))
```

ags_fx_ladspa_channel_processor_get_type ()

```
GType
ags_fx_ladspa_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR                ( ↩
    ags_fx_ladspa_channel_processor_get_type())
```

struct AgsFxLadspaChannelProcessor

```
struct AgsFxLadspaChannelProcessor;
```

struct AgsFxLadspaChannelProcessorClass

```
struct AgsFxLadspaChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

23.6 AgsFxLadspaRecycling

AgsFxLadspaRecycling — fx ladspa recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLadspaRecycling *	ags_fx_ladspa_recycling_new ()
#define	AGS_FX_LADSPA_RECYCLING()
#define	AGS_FX_LADSPA_RECYCLING_CLASS()
#define	AGS_FX_LADSPA_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_LADSPA_RECYCLING()
#define	AGS_IS_FX_LADSPA_RECYCLING_CLASS()
GType	ags_fx_ladspa_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_LADSPA_RECYCLING
struct	AgsFxLadspaRecycling
struct	AgsFxLadspaRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxLadspaRecycling
```

Implemented Interfaces

AgsFxLadspaRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_ladspa_recycling.h>
```

Description

The AgsFxLadspaRecycling class provides ports to the effect processor.

Functions

ags_fx_ladspa_recycling_new ()

```
AgsFxLadspaRecycling~*
ags_fx_ladspa_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxLadspaRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxLadspaRecycling

Since: 3.3.0

AGS_FX_LADSPA_RECYCLING()

```
#define AGS_FX_LADSPA_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LADSPA_RECYCLING, AgsFxLadspaRecycling))
```

AGS_FX_LADSPA_RECYCLING_CLASS()

```
#define AGS_FX_LADSPA_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LADSPA_RECYCLING, AgsFxLadspaRecycling))
```

AGS_FX_LADSPA_RECYCLING_GET_CLASS()

```
#define AGS_FX_LADSPA_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LADSPA_RECYCLING, AgsFxLadspaRecyclingClass))
```

AGS_IS_FX_LADSPA_RECYCLING()

```
#define AGS_IS_FX_LADSPA_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LADSPA_RECYCLING))
```

AGS_IS_FX_LADSPA_RECYCLING_CLASS()

```
#define AGS_IS_FX_LADSPA_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LADSPA_RECYCLING))
```

ags_fx_ladspa_recycling_get_type ()

```
GType
ags_fx_ladspa_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_LADSPA_RECYCLING

```
#define AGS_TYPE_FX_LADSPA_RECYCLING                (ags_fx_ladspa_recycling_get_type())
```

struct AgsFxLadspaRecycling

```
struct AgsFxLadspaRecycling;
```

struct AgsFxLadspaRecyclingClass

```
struct AgsFxLadspaRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

23.7 AgsFxLfoAudio

AgsFxLfoAudio — fx lfo audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLfoAudio *	ags_fx_lfo_audio_new ()
#define	AGS_FX_LFO_AUDIO()
#define	AGS_FX_LFO_AUDIO_CLASS()
#define	AGS_FX_LFO_AUDIO_GET_CLASS()
#define	AGS_IS_FX_LFO_AUDIO()
#define	AGS_IS_FX_LFO_AUDIO_CLASS()
GType	ags_fx_lfo_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_LFO_AUDIO
struct	AgsFxLfoAudio
struct	AgsFxLfoAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxLfoAudio
```

Implemented Interfaces

AgsFxLfoAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lfo_audio.h>
```

Description

The AgsFxLfoAudio class provides ports to the effect processor.

Functions

ags_fx_lfo_audio_new ()

```
AgsFxLfoAudio~*
ags_fx_lfo_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxLfoAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxLfoAudio

Since: 3.8.0

AGS_FX_LFO_AUDIO()

```
#define AGS_FX_LFO_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO, AgsFxLfoAudio))
```

AGS_FX_LFO_AUDIO_CLASS()

```
#define AGS_FX_LFO_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LFO_AUDIO, AgsFxLfoAudioClass))
```

AGS_FX_LFO_AUDIO_GET_CLASS()

```
#define AGS_FX_LFO_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO, AgsFxLfoAudioClass))
```

AGS_IS_FX_LFO_AUDIO()

```
#define AGS_IS_FX_LFO_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO))
```

AGS_IS_FX_LFO_AUDIO_CLASS()

```
#define AGS_IS_FX_LFO_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LFO_AUDIO))
```

ags_fx_lfo_audio_get_type ()

```
GType
ags_fx_lfo_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_LFO_AUDIO

```
#define AGS_TYPE_FX_LFO_AUDIO                (ags_fx_lfo_audio_get_type())
```

struct AgsFxLfoAudio

```
struct AgsFxLfoAudio;
```

struct AgsFxLfoAudioClass

```
struct AgsFxLfoAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

23.8 AgsFxLfoAudioProcessor

AgsFxLfoAudioProcessor — fx lfo audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLfoAudioProcessor *	ags_fx_lfo_audio_processor_new ()
#define	AGS_FX_LFO_AUDIO_PROCESSOR()
#define	AGS_FX_LFO_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_LFO_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LFO_AUDIO_PROCESSOR()
#define	AGS_IS_FX_LFO_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_lfo_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LFO_AUDIO_PROCESSOR
struct	AgsFxLfoAudioProcessor
struct	AgsFxLfoAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxLfoAudioProcessor
```

Implemented Interfaces

AgsFxLfoAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lfo_audio_processor.h>
```

Description

The AgsFxLfoAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_lfo_audio_processor_new ()

```
AgsFxLfoAudioProcessor~*
ags_fx_lfo_audio_processor_new (AgsAudio *audio);
```

Create a new instance of AgsFxLfoAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxLfoAudioProcessor

Since: 3.8.0

AGS_FX_LFO_AUDIO_PROCESSOR()

```
#define AGS_FX_LFO_AUDIO_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO_PROCESSOR, AgsFxLfoAudioProcessor))
```

AGS_FX_LFO_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_LFO_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LFO_AUDIO_PROCESSOR, AgsFxLfoAudioProcessorClass))
```

AGS_FX_LFO_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LFO_AUDIO_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO_PROCESSOR, AgsFxLfoAudioProcessorClass))
```

AGS_IS_FX_LFO_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_LFO_AUDIO_PROCESSOR(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO_PROCESSOR))
```

AGS_IS_FX_LFO_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LFO_AUDIO_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LFO_AUDIO_PROCESSOR))
```

ags_fx_lfo_audio_processor_get_type ()

```
GType
ags_fx_lfo_audio_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_LFO_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_LFO_AUDIO_PROCESSOR                (ags_fx_lfo_audio_processor_get_type ↩
    ())
```

struct AgsFxLfoAudioProcessor

```
struct AgsFxLfoAudioProcessor;
```

struct AgsFxLfoAudioProcessorClass

```
struct AgsFxLfoAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

23.9 AgsFxLfoAudioSignal

AgsFxLfoAudioSignal — fx lfo audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLfoAudioSignal *	ags_fx_lfo_audio_signal_new ()
#define	AGS_FX_LFO_AUDIO_SIGNAL()
#define	AGS_FX_LFO_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_LFO_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_LFO_AUDIO_SIGNAL()
#define	AGS_IS_FX_LFO_AUDIO_SIGNAL_CLASS()
GType	ags_fx_lfo_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_LFO_AUDIO_SIGNAL
struct	AgsFxLfoAudioSignal
struct	AgsFxLfoAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxLfoAudioSignal
```

Implemented Interfaces

AgsFxLfoAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lfo_audio_signal.h>
```

Description

The AgsFxLfoAudioSignal class provides ports to the effect processor.

Functions

ags_fx_lfo_audio_signal_new ()

```
AgsFxLfoAudioSignal~*
ags_fx_lfo_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxLfoAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxLfoAudioSignal

Since: 3.8.0

AGS_FX_LFO_AUDIO_SIGNAL()

```
#define AGS_FX_LFO_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO_SIGNAL, AgsFxLfoAudioSignal))
```

AGS_FX_LFO_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_LFO_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LFO_AUDIO_SIGNAL, AgsFxLfoAudioSignalClass))
```

AGS_FX_LFO_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_LFO_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO_SIGNAL, AgsFxLfoAudioSignalClass))
```

AGS_IS_FX_LFO_AUDIO_SIGNAL()

```
#define AGS_IS_FX_LFO_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LFO_AUDIO_SIGNAL))
```

AGS_IS_FX_LFO_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_LFO_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LFO_AUDIO_SIGNAL))
```

ags_fx_lfo_audio_signal_get_type ()

```
GType
ags_fx_lfo_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_FX_LFO_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_LFO_AUDIO_SIGNAL                (ags_fx_lfo_audio_signal_get_type())
```

struct AgsFxLfoAudioSignal

```
struct AgsFxLfoAudioSignal;
```

struct AgsFxLfoAudioSignalClass

```
struct AgsFxLfoAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

23.10 AgsFxLfoChannel

AgsFxLfoChannel — fx lfo channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLfoChannelInputData *	ags_fx_lfo_channel_input_data_alloc ()
void	ags_fx_lfo_channel_input_data_free ()
AgsFxLfoChannel *	ags_fx_lfo_channel_new ()
#define	AGS_FX_LFO_CHANNEL()
#define	AGS_FX_LFO_CHANNEL_CLASS()
#define	AGS_FX_LFO_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_LFO_CHANNEL()
#define	AGS_IS_FX_LFO_CHANNEL_CLASS()
GType	ags_fx_lfo_channel_get_type ()

Properties

AgsPort *	enabled	Read / Write
AgsPort *	lfo-depth	Read / Write
AgsPort *	lfo-freq	Read / Write
AgsPort *	lfo-phase	Read / Write
AgsPort *	lfo-tuning	Read / Write
AgsPort *	lfo-wave	Read / Write

Types and Values

struct	AgsFxLfoChannelInputData
#define	AGS_TYPE_FX_LFO_CHANNEL
struct	AgsFxLfoChannel
struct	AgsFxLfoChannelClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannel
        &#x2570;&#x2500;&#x2500; AgsFxLfoChannel
```

Implemented Interfaces

AgsFxLfoChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lfo_channel.h>
```

Description

The AgsFxLfoChannel class provides ports to the effect processor.

Functions

ags_fx_lfo_channel_input_data_alloc ()

```
AgsFxLfoChannelInputData~*
ags_fx_lfo_channel_input_data_alloc ();
```

Allocate AgsFxLfoChannelInputData

Returns

the new AgsFxLfoChannelInputData

Since: 4.0.0

ags_fx_lfo_channel_input_data_free ()

```
void
ags_fx_lfo_channel_input_data_free (AgsFxLfoChannelInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxLfoChannelInputData

Since: 3.3.0

ags_fx_lfo_channel_new ()

```
AgsFxLfoChannel~*
ags_fx_lfo_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxLfoChannel

Parameters

channel	the AgsChannel

Returns

the new AgsFxLfoChannel

Since: 3.8.0

AGS_FX_LFO_CHANNEL()

```
#define AGS_FX_LFO_CHANNEL(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LFO_CHANNEL, AgsFxLfoChannel))
```

AGS_FX_LFO_CHANNEL_CLASS()

```
#define AGS_FX_LFO_CHANNEL_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LFO_CHANNEL, AgsFxLfoChannelClass))
```

AGS_FX_LFO_CHANNEL_GET_CLASS()

```
#define AGS_FX_LFO_CHANNEL_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LFO_CHANNEL, AgsFxLfoChannelClass))
```

AGS_IS_FX_LFO_CHANNEL()

```
#define AGS_IS_FX_LFO_CHANNEL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LFO_CHANNEL))
```

AGS_IS_FX_LFO_CHANNEL_CLASS()

```
#define AGS_IS_FX_LFO_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LFO_CHANNEL))
```

ags_fx_lfo_channel_get_type ()

```
GType
ags_fx_lfo_channel_get_type ();
```

Types and Values

struct AgsFxLfoChannelInputData

```
struct AgsFxLfoChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  AgsLFOSynthUtil *lfo_synth_util;
};
```

AGS_TYPE_FX_LFO_CHANNEL

```
#define AGS_TYPE_FX_LFO_CHANNEL                 (ags_fx_lfo_channel_get_type())
```

struct AgsFxLfoChannel

```
struct AgsFxLfoChannel;
```

struct AgsFxLfoChannelClass

```
struct AgsFxLfoChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "enabled" property

```
  "enabled"                   AgsPort~*
```

The enable control of LFO.

Owner: AgsFxLfoChannel

Flags: Read / Write

Since: 3.8.0

The "lfo-depth" property

```
  "lfo-depth"                 AgsPort~*
```

The depth of LFO.

Owner: AgsFxLfoChannel

Flags: Read / Write

Since: 3.8.0

The "lfo-freq" property

```
  "lfo-freq"                  AgsPort~*
```

The freq of LFO.

Owner: AgsFxLfoChannel

Flags: Read / Write

Since: 3.8.0

The "lfo-phase" property

```
  "lfo-phase"                 AgsPort~*
```

The phase of LFO.

Owner: AgsFxLfoChannel

Flags: Read / Write

Since: 3.8.0

The "lfo-tuning" property

```
"lfo-tuning"                AgsPort~*
```

The tuning of LFO.

Owner: AgsFxLfoChannel

Flags: Read / Write

Since: 3.8.0

The "lfo-wave" property

```
"lfo-wave"                  AgsPort~*
```

The wave of LFO.

Owner: AgsFxLfoChannel

Flags: Read / Write

Since: 3.8.0

23.11 AgsFxLfoChannelProcessor

AgsFxLfoChannelProcessor — fx lfo channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLfoChannelProcessor *	ags_fx_lfo_channel_processor_new ()
#define	AGS_FX_LFO_CHANNEL_PROCESSOR()
#define	AGS_FX_LFO_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_LFO_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_LFO_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_LFO_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_lfo_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR
struct	AgsFxLfoChannelProcessor
struct	AgsFxLfoChannelProcessorClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsRecall
    &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
        &#x2570;&#x2500;&#x2500; AgsFxLfoChannelProcessor
```

Implemented Interfaces

AgsFxLfoChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lfo_channel_processor.h>
```

Description

The AgsFxLfoChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_lfo_channel_processor_new ()

```
AgsFxLfoChannelProcessor~*
ags_fx_lfo_channel_processor_new (AgsChannel *channel);
```

Create a new instance of AgsFxLfoChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxLfoChannelProcessor

Since: 3.3.0

AGS_FX_LFO_CHANNEL_PROCESSOR()

```
#define AGS_FX_LFO_CHANNEL_PROCESSOR(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR, AgsFxLfoChannelProcessor))
```

AGS_FX_LFO_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_LFO_CHANNEL_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR, AgsFxLfoChannelProcessorClass))
```

AGS_FX_LFO_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_LFO_CHANNEL_PROCESSOR_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
    AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR, AgsFxLfoChannelProcessorClass))
```

AGS_IS_FX_LFO_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_LFO_CHANNEL_PROCESSOR(obj)            (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR))
```

AGS_IS_FX_LFO_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_LFO_CHANNEL_PROCESSOR_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR))
```

ags_fx_lfo_channel_processor_get_type ()

```
GType
ags_fx_lfo_channel_processor_get_type ();
```

Types and Values

AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR                ( ↩
    ags_fx_lfo_channel_processor_get_type())
```

struct AgsFxLfoChannelProcessor

```
struct AgsFxLfoChannelProcessor;
```

struct AgsFxLfoChannelProcessorClass

```
struct AgsFxLfoChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

23.12 AgsFxLfoRecycling

AgsFxLfoRecycling — fx lfo recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxLfoRecycling *	ags_fx_lfo_recycling_new ()
#define	AGS_FX_LFO_RECYCLING()
#define	AGS_FX_LFO_RECYCLING_CLASS()
#define	AGS_FX_LFO_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_LFO_RECYCLING()
#define	AGS_IS_FX_LFO_RECYCLING_CLASS()
GType	ags_fx_lfo_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_LFO_RECYCLING
struct	AgsFxLfoRecycling
struct	AgsFxLfoRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxLfoRecycling
```

Implemented Interfaces

AgsFxLfoRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_lfo_recycling.h>
```

Description

The AgsFxLfoRecycling class provides ports to the effect processor.

Functions

ags_fx_lfo_recycling_new ()

```
AgsFxLfoRecycling~*
ags_fx_lfo_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxLfoRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxLfoRecycling

Since: 3.8.0

AGS_FX_LFO_RECYCLING()

```
#define AGS_FX_LFO_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FX_LFO_RECYCLING, AgsFxLfoRecycling))
```

AGS_FX_LFO_RECYCLING_CLASS()

```
#define AGS_FX_LFO_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_LFO_RECYCLING, AgsFxLfoRecyclingClass))
```

AGS_FX_LFO_RECYCLING_GET_CLASS()

```
#define AGS_FX_LFO_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FX_LFO_RECYCLING, AgsFxLfoRecyclingClass))
```

AGS_IS_FX_LFO_RECYCLING()

```
#define AGS_IS_FX_LFO_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FX_LFO_RECYCLING))
```

AGS_IS_FX_LFO_RECYCLING_CLASS()

```
#define AGS_IS_FX_LFO_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FX_LFO_RECYCLING))
```

ags_fx_lfo_recycling_get_type ()

```
GType
ags_fx_lfo_recycling_get_type ();
```

Types and Values

AGS_TYPE_FX_LFO_RECYCLING

```
#define AGS_TYPE_FX_LFO_RECYCLING                (ags_fx_lfo_recycling_get_type())
```

struct AgsFxLfoRecycling

```
struct AgsFxLfoRecycling;
```

struct AgsFxLfoRecyclingClass

```
struct AgsFxLfoRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

23.13 AgsFxTwoPassAliaseAudio

AgsFxTwoPassAliaseAudio — fx two pass aliase audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxTwoPassAliaseAudio *	ags_fx_two_pass_aliase_audio_new ()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_CLASS()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_GET_CLASS()
#define	AGS_IS_FX_TWO_PASS_ALIASE_AUDIO()
#define	AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_CLASS()
GType	ags_fx_two_pass_aliase_audio_get_type ()

Types and Values

#define	AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO
struct	AgsFxTwoPassAliaseAudio
struct	AgsFxTwoPassAliaseAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudio
            &#x2570;&#x2500;&#x2500; AgsFxTwoPassAliaseAudio
```

Implemented Interfaces

AgsFxTwoPassAliaseAudio implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_two_pass_aliase_audio.h>
```

Description

The AgsFxTwoPassAliaseAudio class provides ports to the effect processor.

Functions

ags_fx_two_pass_aliase_audio_new ()

```
AgsFxTwoPassAliaseAudio~*
ags_fx_two_pass_aliase_audio_new (AgsAudio *audio);
```

Create a new instance of AgsFxTwoPassAliaseAudio

Parameters

audio	the AgsAudio

Returns

the new AgsFxTwoPassAliaseAudio

Since: 3.8.0

AGS_FX_TWO_PASS_ALIASE_AUDIO()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
     AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO, AgsFxTwoPassAliaseAudio))
```

AGS_FX_TWO_PASS_ALIASE_AUDIO_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO, AgsFxTwoPassAliaseAudioClass))
```

AGS_FX_TWO_PASS_ALIASE_AUDIO_GET_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj), ↩
     AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO, AgsFxTwoPassAliaseAudioClass))
```

AGS_IS_FX_TWO_PASS_ALIASE_AUDIO()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj) ↩
    , AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO))
```

AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_CLASS()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), ↩
     AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO))
```

ags_fx_two_pass_aliase_audio_get_type ()

```
GType
ags_fx_two_pass_aliase_audio_get_type ();
```

Types and Values

AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO

```
#define AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO                ( ↩
    ags_fx_two_pass_aliase_audio_get_type())
```

struct AgsFxTwoPassAliaseAudio

```
struct AgsFxTwoPassAliaseAudio;
```

struct AgsFxTwoPassAliaseAudioClass

```
struct AgsFxTwoPassAliaseAudioClass {
  AgsRecallAudioClass recall_audio;
};
```

23.14 AgsFxTwoPassAliaseAudioProcessor

AgsFxTwoPassAliaseAudioProcessor — fx two pass aliase audio processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxTwoPassAliaseAudioProcessor *	ags_fx_two_pass_aliase_audio_processor_new ()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR()
#define	AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS()
GType	ags_fx_two_pass_aliase_audio_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR
struct	AgsFxTwoPassAliaseAudioProcessor
struct	AgsFxTwoPassAliaseAudioProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioRun
            &#x2570;&#x2500;&#x2500; AgsFxTwoPassAliaseAudioProcessor
```

Implemented Interfaces

AgsFxTwoPassAliaseAudioProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_two_pass_aliase_audio_processor.h>
```

Description

The AgsFxTwoPassAliaseAudioProcessor class provides ports to the effect processor.

Functions

ags_fx_two_pass_aliase_audio_processor_new ()

```
AgsFxTwoPassAliaseAudioProcessor~*
ags_fx_two_pass_aliase_audio_processor_new
                                (AgsAudio *audio);
```

Create a new instance of AgsFxTwoPassAliaseAudioProcessor

Parameters

audio	the AgsAudio

Returns

the new AgsFxTwoPassAliaseAudioProcessor

Since: 3.8.0

AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR(obj)                ( ↩
    G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR,  ↩
    AgsFxTwoPassAliaseAudioProcessor))
```

AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST ↩
    ((class), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR,  ↩
    AgsFxTwoPassAliaseAudioProcessorClass))
```

AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_GET_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_GET_CLASS(obj)      ( ↩
    G_TYPE_INSTANCE_GET_CLASS ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR,  ↩
    AgsFxTwoPassAliaseAudioProcessorClass))
```

AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR(obj)              ( ↩
    G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR))
```

AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ↩
     ((class), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR))
```

ags_fx_two_pass_aliase_audio_processor_get_type ()

```
GType
ags_fx_two_pass_aliase_audio_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR

```
#define AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR                 ( ↩
    ags_fx_two_pass_aliase_audio_processor_get_type())
```

struct AgsFxTwoPassAliaseAudioProcessor

```
struct AgsFxTwoPassAliaseAudioProcessor;
```

struct AgsFxTwoPassAliaseAudioProcessorClass

```
struct AgsFxTwoPassAliaseAudioProcessorClass {
  AgsRecallAudioRunClass recall_audio_run;
};
```

23.15 AgsFxTwoPassAliaseAudioSignal

AgsFxTwoPassAliaseAudioSignal — fx two pass aliase audio signal

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxTwoPassAliaseAudioSignal *	ags_fx_two_pass_aliase_audio_signal_new ()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS()
#define	AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL()
#define	AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS()
GType	ags_fx_two_pass_aliase_audio_signal_get_type ()

Types and Values

#define	AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL
struct	AgsFxTwoPassAliaseAudioSignal
struct	AgsFxTwoPassAliaseAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallAudioSignal
            &#x2570;&#x2500;&#x2500; AgsFxTwoPassAliaseAudioSignal
```

Implemented Interfaces

AgsFxTwoPassAliaseAudioSignal implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_two_pass_aliase_audio_signal.h>
```

Description

The AgsFxTwoPassAliaseAudioSignal class provides ports to the effect processor.

Functions

ags_fx_two_pass_aliase_audio_signal_new ()

```
AgsFxTwoPassAliaseAudioSignal~*
ags_fx_two_pass_aliase_audio_signal_new
                                (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsFxTwoPassAliaseAudioSignal

Parameters

audio_signal	the AgsAudioSignal

Returns

the new AgsFxTwoPassAliaseAudioSignal

Since: 3.8.0

AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST ↩
    ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL, AgsFxTwoPassAliaseAudioSignal))
```

AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL, AgsFxTwoPassAliaseAudioSignalClass))
```

AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS  ↩
    ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL, AgsFxTwoPassAliaseAudioSignalClass))
```

AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ↩
     ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL))
```

AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL))
```

ags_fx_two_pass_aliase_audio_signal_get_type ()

```
GType
ags_fx_two_pass_aliase_audio_signal_get_type
                                    ();
```

Types and Values

AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL

```
#define AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL                ( ↩
    ags_fx_two_pass_aliase_audio_signal_get_type())
```

struct AgsFxTwoPassAliaseAudioSignal

```
struct AgsFxTwoPassAliaseAudioSignal;
```

struct AgsFxTwoPassAliaseAudioSignalClass

```
struct AgsFxTwoPassAliaseAudioSignalClass {
  AgsRecallAudioSignalClass recall_audio_signal;
};
```

23.16 AgsFxTwoPassAliaseChannel

AgsFxTwoPassAliaseChannel — fx two pass aliase channel

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()
AgsFxTwoPassAliaseChannelInputData *	ags_fx_two_pass_aliase_channel_input_data_alloc ()
void	ags_fx_two_pass_aliase_channel_input_data_free ()
GRecMutex *	ags_fx_two_pass_aliase_channel_input_data_get_strct_mutex ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_parent ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_orig_buffer ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_a_buffer ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_b_buffer ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_a_mix ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_b_mix ()
gpointer	ags_fx_two_pass_aliase_channel_input_get_final_mix ()
AgsFxTwoPassAliaseChannel *	ags_fx_two_pass_aliase_channel_new ()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_GET_CLASS()
#define	AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL()
#define	AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS()
GType	ags_fx_two_pass_aliase_channel_get_type ()

Properties

AgsPort *	a-amount	Read / Write
AgsPort *	a-phase	Read / Write
AgsPort *	b-amount	Read / Write
AgsPort *	b-phase	Read / Write
AgsPort *	enabled	Read / Write

Types and Values

struct	AgsFxTwoPassAliaseChannelInputData
#define	AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL
struct	AgsFxTwoPassAliaseChannel

struct	AgsFxTwoPassAliaseChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannel
            &#x2570;&#x2500;&#x2500; AgsFxTwoPassAliaseChannel
```

Implemented Interfaces

AgsFxTwoPassAliaseChannel implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_two_pass_aliase_channel.h>
```

Description

The AgsFxTwoPassAliaseChannel class provides ports to the effect processor.

Functions

AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA(ptr) ((AgsFxTwoPassAliaseChannelInputData ↩
     *)(ptr))
```

AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX(ptr) (&((( ↩
    AgsFxTwoPassAliaseChannelInputData *)(ptr))->strct_mutex))
```

ags_fx_two_pass_aliase_channel_input_data_alloc ()

```
AgsFxTwoPassAliaseChannelInputData~*
ags_fx_two_pass_aliase_channel_input_data_alloc
                                ();
```

Allocate AgsFxTwoPassAliaseChannelInputData

Returns

the new AgsFxTwoPassAliaseChannelInputData.

[type gpointer][transfer full]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_data_free ()

```
void
ags_fx_two_pass_aliase_channel_input_data_free
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Free *input_data* .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer full]*

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_data_get_strct_mutex ()

```
GRecMutex~*
ags_fx_two_pass_aliase_channel_input_data_get_strct_mutex
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get structure mutex.

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the GRecMutex to lock *input_data* .

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_parent ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_parent
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get parent of *input_data* .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the parent.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_orig_buffer ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_orig_buffer
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get orig buffer of *input_data* .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the orig buffer.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_a_buffer ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_a_buffer
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get a buffer of *input_data* .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the a buffer.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_b_buffer ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_b_buffer
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get b buffer of *input_data* .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the b buffer.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_a_mix ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_a_mix
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get a mix of `input_data` .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the a mix.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_b_mix ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_b_mix
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get b mix of `input_data` .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the b mix.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_input_get_final_mix ()

```
gpointer
ags_fx_two_pass_aliase_channel_input_get_final_mix
                                (AgsFxTwoPassAliaseChannelInputData *input_data);
```

Get final mix of *input_data* .

Parameters

input_data	the AgsFxTwoPassAliaseChannelInputData.	*[type gpointer][transfer none]*

Returns

the final mix.

[type gpointer][transfer none]

Since: 3.8.0

ags_fx_two_pass_aliase_channel_new ()

```
AgsFxTwoPassAliaseChannel~*
ags_fx_two_pass_aliase_channel_new (AgsChannel *channel);
```

Create a new instance of AgsFxTwoPassAliaseChannel

Parameters

channel	the AgsChannel	

Returns

the new AgsFxTwoPassAliaseChannel

Since: 3.8.0

AGS_FX_TWO_PASS_ALIASE_CHANNEL()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL, AgsFxTwoPassAliaseChannel))
```

AGS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class) ↩
    , AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL, AgsFxTwoPassAliaseChannelClass))
```

AGS_FX_TWO_PASS_ALIASE_CHANNEL_GET_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj ↩
    ), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL, AgsFxTwoPassAliaseChannelClass))
```

AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL))
```

AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL))
```

ags_fx_two_pass_aliase_channel_get_type ()

```
GType
ags_fx_two_pass_aliase_channel_get_type
                                ();
```

Types and Values

struct AgsFxTwoPassAliaseChannelInputData

```
struct AgsFxTwoPassAliaseChannelInputData {
  GRecMutex strct_mutex;

  gpointer parent;

  gdouble *orig_buffer;

  gdouble *a_buffer;
  gdouble *b_buffer;

  gdouble *a_mix;
  gdouble *b_mix;

  gdouble *final_mix;
};
```

AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL

```
#define AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL                 ( ↩
    ags_fx_two_pass_aliase_channel_get_type())
```

struct AgsFxTwoPassAliaseChannel

```
struct AgsFxTwoPassAliaseChannel;
```

struct AgsFxTwoPassAliaseChannelClass

```
struct AgsFxTwoPassAliaseChannelClass {
  AgsRecallChannelClass recall_channel;
};
```

Property Details

The "a-amount" property

```
"a-amount"                  AgsPort~*
```

The first pass amount.

Owner: AgsFxTwoPassAliaseChannel

Flags: Read / Write

Since: 3.8.0

The "a-phase" property

```
"a-phase"                   AgsPort~*
```

The first pass phase.

Owner: AgsFxTwoPassAliaseChannel

Flags: Read / Write

Since: 3.8.0

The "b-amount" property

```
"b-amount"                  AgsPort~*
```

The second pass amount.

Owner: AgsFxTwoPassAliaseChannel

Flags: Read / Write

Since: 3.8.0

The "b-phase" property

```
"b-phase"                   AgsPort~*
```

The second pass phase.

Owner: AgsFxTwoPassAliaseChannel

Flags: Read / Write

Since: 3.8.0

The "enabled" property

```
"enabled"                   AgsPort~*
```

The enable control of two pass aliase.

Owner: AgsFxTwoPassAliaseChannel

Flags: Read / Write

Since: 3.8.0

23.17 AgsFxTwoPassAliaseChannelProcessor

AgsFxTwoPassAliaseChannelProcessor — fx two pass aliase channel processor

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxTwoPassAliaseChannelProcessor *	ags_fx_two_pass_aliase_channel_processor_new ()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS()
#define	AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_GET_CLASS()
#define	AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR()
#define	AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS()
GType	ags_fx_two_pass_aliase_channel_processor_get_type ()

Types and Values

#define	AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR
struct	AgsFxTwoPassAliaseChannelProcessor
struct	AgsFxTwoPassAliaseChannelProcessorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallChannelRun
            &#x2570;&#x2500;&#x2500; AgsFxTwoPassAliaseChannelProcessor
```

Implemented Interfaces

AgsFxTwoPassAliaseChannelProcessor implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_two_pass_aliase_channel_processor.h>
```

Description

The AgsFxTwoPassAliaseChannelProcessor class provides ports to the effect processor.

Functions

ags_fx_two_pass_aliase_channel_processor_new ()

```
AgsFxTwoPassAliaseChannelProcessor~*
ags_fx_two_pass_aliase_channel_processor_new
                                (AgsChannel *channel);
```

Create a new instance of AgsFxTwoPassAliaseChannelProcessor

Parameters

channel	the AgsChannel

Returns

the new AgsFxTwoPassAliaseChannelProcessor

Since: 3.8.0

AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR(obj)                ( ↩
    G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR,  ↩
    AgsFxTwoPassAliaseChannelProcessor))
```

AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS(class)        ( ↩
    G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR,  ↩
    AgsFxTwoPassAliaseChannelProcessorClass))
```

AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_GET_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_GET_CLASS(obj)      ( ↩
    G_TYPE_INSTANCE_GET_CLASS ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR,  ↩
    AgsFxTwoPassAliaseChannelProcessorClass))
```

AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR(obj)             ( ↩
    G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR))
```

AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS(class)     ( ↩
    G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR))
```

ags_fx_two_pass_aliase_channel_processor_get_type ()

```
GType
ags_fx_two_pass_aliase_channel_processor_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR

```
#define AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR                ( ↩
    ags_fx_two_pass_aliase_channel_processor_get_type())
```

struct AgsFxTwoPassAliaseChannelProcessor

```
struct AgsFxTwoPassAliaseChannelProcessor;
```

struct AgsFxTwoPassAliaseChannelProcessorClass

```
struct AgsFxTwoPassAliaseChannelProcessorClass {
  AgsRecallChannelRunClass recall_channel_run;
};
```

23.18 AgsFxTwoPassAliaseRecycling

AgsFxTwoPassAliaseRecycling — fx two pass aliase recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsFxTwoPassAliaseRecycling *	ags_fx_two_pass_aliase_recycling_new ()
#define	AGS_FX_TWO_PASS_ALIASE_RECYCLING()
#define	AGS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS()
#define	AGS_FX_TWO_PASS_ALIASE_RECYCLING_GET_CLASS()
#define	AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING()
#define	AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS()
GType	ags_fx_two_pass_aliase_recycling_get_type ()

Types and Values

#define	AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING
struct	AgsFxTwoPassAliaseRecycling
struct	AgsFxTwoPassAliaseRecyclingClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRecall
        &#x2570;&#x2500;&#x2500; AgsRecallRecycling
            &#x2570;&#x2500;&#x2500; AgsFxTwoPassAliaseRecycling
```

Implemented Interfaces

AgsFxTwoPassAliaseRecycling implements AgsConnectable.

Includes

```
#include <ags/audio/fx/ags_fx_two_pass_aliase_recycling.h>
```

Description

The AgsFxTwoPassAliaseRecycling class provides ports to the effect processor.

Functions

ags_fx_two_pass_aliase_recycling_new ()

```
AgsFxTwoPassAliaseRecycling~*
ags_fx_two_pass_aliase_recycling_new (AgsRecycling *recycling);
```

Create a new instance of AgsFxTwoPassAliaseRecycling

Parameters

recycling	the AgsRecycling

Returns

the new AgsFxTwoPassAliaseRecycling

Since: 3.8.0

AGS_FX_TWO_PASS_ALIASE_RECYCLING()

```
#define AGS_FX_TWO_PASS_ALIASE_RECYCLING(obj)                (G_TYPE_CHECK_INSTANCE_CAST(( ↩
    obj), AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING, AgsFxTwoPassAliaseRecycling))
```

AGS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(( ↩
    class), AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING, AgsFxTwoPassAliaseRecyclingClass))
```

AGS_FX_TWO_PASS_ALIASE_RECYCLING_GET_CLASS()

```
#define AGS_FX_TWO_PASS_ALIASE_RECYCLING_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (( ↩
    obj), AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING, AgsFxTwoPassAliaseRecyclingClass))
```

AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING))
```

AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS()

```
#define AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE (( ↩
    class), AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING))
```

ags_fx_two_pass_aliase_recycling_get_type ()

```
GType
ags_fx_two_pass_aliase_recycling_get_type
                                ();
```

Types and Values

AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING

```
#define AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING                ( ↩
    ags_fx_two_pass_aliase_recycling_get_type())
```

struct AgsFxTwoPassAliaseRecycling

```
struct AgsFxTwoPassAliaseRecycling;
```

struct AgsFxTwoPassAliaseRecyclingClass

```
struct AgsFxTwoPassAliaseRecyclingClass {
  AgsRecallRecyclingClass recall_recycling;
};
```

Chapter 24

Audio tasks - Data exchanged between Audio and GUI

Tasks ensure to be run conflict free asynchrononous exclusively.

24.1 AgsAddAudio

AgsAddAudio — add audio object to application context

Stability Level

Stable, unless otherwise indicated

Functions

AgsAddAudio *	ags_add_audio_new ()
#define	AGS_ADD_AUDIO()
#define	AGS_ADD_AUDIO_CLASS()
#define	AGS_ADD_AUDIO_GET_CLASS()
#define	AGS_IS_ADD_AUDIO()
#define	AGS_IS_ADD_AUDIO_CLASS()
GType	ags_add_audio_get_type ()

Properties

AgsAudio *	audio	Read / Write

Types and Values

#define	AGS_TYPE_ADD_AUDIO
struct	AgsAddAudio
struct	AgsAddAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsAddAudio
```

Implemented Interfaces

AgsAddAudio implements AgsConnectable.

Includes

```
#include <ags/audio/task/ags_add_audio.h>
```

Description

The AgsAddAudio task adds AgsAudio to AgsApplicationContext.

Functions

ags_add_audio_new ()

```
AgsAddAudio~*
ags_add_audio_new (AgsAudio *audio);
```

Creates an AgsAddAudio.

Parameters

audio	the AgsAudio to add

Returns

an new AgsAddAudio.

Since: 3.0.0

AGS_ADD_AUDIO()

```
#define AGS_ADD_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ADD_AUDIO, AgsAddAudio))
```

AGS_ADD_AUDIO_CLASS()

```
#define AGS_ADD_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_ADD_AUDIO, AgsAddAudioClass))
```

AGS_ADD_AUDIO_GET_CLASS()

```
#define AGS_ADD_AUDIO_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_ADD_AUDIO, AgsAddAudioClass))
```

AGS_IS_ADD_AUDIO()

```
#define AGS_IS_ADD_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_ADD_AUDIO))
```

AGS_IS_ADD_AUDIO_CLASS()

```
#define AGS_IS_ADD_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_ADD_AUDIO))
```

ags_add_audio_get_type ()

```
GType
ags_add_audio_get_type ();
```

Types and Values

AGS_TYPE_ADD_AUDIO

```
#define AGS_TYPE_ADD_AUDIO                 (ags_add_audio_get_type())
```

struct AgsAddAudio

```
struct AgsAddAudio;
```

struct AgsAddAudioClass

```
struct AgsAddAudioClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsAddAudio

Flags: Read / Write

Since: 3.0.0

24.2 AgsAddAudioSignal

AgsAddAudioSignal — add audio signal object to recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsAddAudioSignal *	ags_add_audio_signal_new ()
#define	AGS_ADD_AUDIO_SIGNAL()
#define	AGS_ADD_AUDIO_SIGNAL_CLASS()
#define	AGS_ADD_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_ADD_AUDIO_SIGNAL()
#define	AGS_IS_ADD_AUDIO_SIGNAL_CLASS()
GType	ags_add_audio_signal_get_type ()

Properties

AgsAudioSignal *	audio-signal	Read / Write
guint	audio-signal-flags	Read / Write
AgsRecallID *	recall-id	Read / Write
AgsRecycling *	recycling	Read / Write
GObject *	soundcard	Read / Write

Types and Values

#define	AGS_TYPE_ADD_AUDIO_SIGNAL
struct	AgsAddAudioSignal
struct	AgsAddAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsAddAudioSignal
```

Includes

```
#include <ags/audio/task/ags_add_audio_signal.h>
```

Description

The AgsAddAudioSignal task adds AgsAudioSignal to AgsRecycling.

Functions

ags_add_audio_signal_new ()

```
AgsAddAudioSignal~*
ags_add_audio_signal_new (AgsRecycling *recycling,
                          AgsAudioSignal *audio_signal,
                          GObject *soundcard,
                          AgsRecallID *recall_id,
                          guint audio_signal_flags);
```

Creates an AgsAddAudioSignal.

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal to add	
soundcard	the GObject defaulting to	
recall_id	the AgsRecallID, may be NULL if AGS_AUDIO_SIGNAL_TEMPLATE set	
audio_signal_flags	the flags to set	

Returns

an new AgsAddAudioSignal.

Since: 3.0.0

AGS_ADD_AUDIO_SIGNAL()

```
#define AGS_ADD_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ADD_AUDIO_SIGNAL, AgsAddAudioSignal))
```

AGS_ADD_AUDIO_SIGNAL_CLASS()

```
#define AGS_ADD_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_ADD_AUDIO_SIGNAL, AgsAddAudioSignalClass))
```

AGS_ADD_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_ADD_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_ADD_AUDIO_SIGNAL, AgsAddAudioSignalClass))
```

AGS_IS_ADD_AUDIO_SIGNAL()

```
#define AGS_IS_ADD_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_ADD_AUDIO_SIGNAL))
```

AGS_IS_ADD_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_ADD_AUDIO_SIGNAL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_ADD_AUDIO_SIGNAL))
```

ags_add_audio_signal_get_type ()

```
GType
ags_add_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_ADD_AUDIO_SIGNAL

```
#define AGS_TYPE_ADD_AUDIO_SIGNAL                 (ags_add_audio_signal_get_type())
```

struct AgsAddAudioSignal

```
struct AgsAddAudioSignal;
```

struct AgsAddAudioSignalClass

```
struct AgsAddAudioSignalClass {
  AgsTaskClass task;
};
```

Property Details

The "audio-signal" property

```
  "audio-signal"              AgsAudioSignal~*
```

The assigned AgsAudioSignal

Owner: AgsAddAudioSignal

Flags: Read / Write

Since: 3.0.0

The "audio-signal-flags" property

```
  "audio-signal-flags"        guint
```

The audio signal's flags.

Owner: AgsAddAudioSignal

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "recall-id" property

```
"recall-id"                 AgsRecallID~*
```

The assigned AgsRecallID

Owner: AgsAddAudioSignal

Flags: Read / Write

Since: 3.0.0

The "recycling" property

```
"recycling"                 AgsRecycling~*
```

The assigned AgsRecycling

Owner: AgsAddAudioSignal

Flags: Read / Write

Since: 3.0.0

The "soundcard" property

```
"soundcard"                 GObject~*
```

The assigned AgsSoundcard

Owner: AgsAddAudioSignal

Flags: Read / Write

Since: 3.0.0

24.3 AgsAddNote

AgsAddNote — add note object to notation

Stability Level

Stable, unless otherwise indicated

Functions

AgsAddNote *	ags_add_note_new ()
#define	AGS_ADD_NOTE()
#define	AGS_ADD_NOTE_CLASS()
#define	AGS_ADD_NOTE_GET_CLASS()
#define	AGS_IS_ADD_NOTE()
#define	AGS_IS_ADD_NOTE_CLASS()
GType	ags_add_note_get_type ()

Properties

AgsAudio *	audio	Read / Write
guint	audio-channel	Read / Write
AgsNote *	note	Read / Write
gboolean	use-selection-list	Read / Write

Types and Values

#define	AGS_TYPE_ADD_NOTE
struct	AgsAddNote
struct	AgsAddNoteClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsAddNote
```

Includes

```
#include <ags/audio/task/ags_add_note.h>
```

Description

The AgsAddNote task adds AgsNote to AgsNotation.

Functions

ags_add_note_new ()

```
AgsAddNote~*
ags_add_note_new (AgsAudio *audio,
                  AgsNote *note,
                  guint audio_channel,
                  gboolean use_selection_list);
```

Create a new instance of AgsAddNote.

Parameters

audio	the AgsAudio	
note	the AgsNote to add	
audio_channel	the audio channel	
use_selection_list	if TRUE added to selection, otherwise to notation	

Returns

the new AgsAddNote

Since: 3.0.0

AGS_ADD_NOTE()

```
#define AGS_ADD_NOTE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ADD_NOTE, AgsAddNote))
```

AGS_ADD_NOTE_CLASS()

```
#define AGS_ADD_NOTE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_ADD_NOTE, AgsAddNoteClass))
```

AGS_ADD_NOTE_GET_CLASS()

```
#define AGS_ADD_NOTE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_ADD_NOTE, AgsAddNoteClass))
```

AGS_IS_ADD_NOTE()

```
#define AGS_IS_ADD_NOTE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_ADD_NOTE))
```

AGS_IS_ADD_NOTE_CLASS()

```
#define AGS_IS_ADD_NOTE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_ADD_NOTE))
```

ags_add_note_get_type ()

```
GType
ags_add_note_get_type ();
```

Types and Values

AGS_TYPE_ADD_NOTE

```
#define AGS_TYPE_ADD_NOTE                (ags_add_note_get_type())
```

struct AgsAddNote

```
struct AgsAddNote;
```

struct AgsAddNoteClass

```
struct AgsAddNoteClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
"audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsAddNote

Flags: Read / Write

Since: 3.0.0

The "audio-channel" property

```
"audio-channel"            guint
```

The assigned audio channel

Owner: AgsAddNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "note" property

```
"note"                     AgsNote~*
```

The assigned AgsNote

Owner: AgsAddNote

Flags: Read / Write

Since: 3.0.0

The "use-selection-list" property

```
"use-selection-list"       gboolean
```

The notation's use-selection-list.

Owner: AgsAddNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

24.4 AgsAddSoundcard

AgsAddSoundcard — add soundcard object to context

Stability Level

Stable, unless otherwise indicated

Functions

AgsAddSoundcard *	ags_add_soundcard_new ()
#define	AGS_ADD_SOUNDCARD()
#define	AGS_ADD_SOUNDCARD_CLASS()
#define	AGS_ADD_SOUNDCARD_GET_CLASS()
#define	AGS_IS_ADD_SOUNDCARD()
#define	AGS_IS_ADD_SOUNDCARD_CLASS()
GType	ags_add_soundcard_get_type ()

Properties

GObject *	soundcard	Read / Write

Types and Values

#define	AGS_TYPE_ADD_SOUNDCARD
struct	AgsAddSoundcard
struct	AgsAddSoundcardClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsAddSoundcard
```

Includes

```
#include <ags/audio/task/ags_add_soundcard.h>
```

Description

The AgsAddSoundcard task adds AgsSoundcard to context.

Functions

ags_add_soundcard_new ()

```
AgsAddSoundcard~*
ags_add_soundcard_new (GObject *soundcard);
```

Create a new instance of AgsAddSoundcard.

Parameters

soundcard	the AgsSoundcard to add

Returns

the new AgsAddSoundcard.

Since: 3.0.0

AGS_ADD_SOUNDCARD()

```
#define AGS_ADD_SOUNDCARD(obj)                  (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_ADD_SOUNDCARD, AgsAddSoundcard))
```

AGS_ADD_SOUNDCARD_CLASS()

```
#define AGS_ADD_SOUNDCARD_CLASS(class)          (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_ADD_SOUNDCARD, AgsAddSoundcardClass))
```

AGS_ADD_SOUNDCARD_GET_CLASS()

```
#define AGS_ADD_SOUNDCARD_GET_CLASS(obj)        (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_ADD_SOUNDCARD, AgsAddSoundcardClass))
```

AGS_IS_ADD_SOUNDCARD()

```
#define AGS_IS_ADD_SOUNDCARD(obj)               (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_ADD_SOUNDCARD))
```

AGS_IS_ADD_SOUNDCARD_CLASS()

```
#define AGS_IS_ADD_SOUNDCARD_CLASS(class)       (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_ADD_SOUNDCARD))
```

ags_add_soundcard_get_type ()

```
GType
ags_add_soundcard_get_type ();
```

Types and Values

AGS_TYPE_ADD_SOUNDCARD

```
#define AGS_TYPE_ADD_SOUNDCARD                  (ags_add_soundcard_get_type())
```

struct AgsAddSoundcard

```
struct AgsAddSoundcard;
```

struct AgsAddSoundcardClass

```
struct AgsAddSoundcardClass {
  AgsTaskClass task;
};
```

Property Details

The "soundcard" property

```
  "soundcard"                GObject~*
```

The assigned AgsSoundcard

Owner: AgsAddSoundcard

Flags: Read / Write

Since: 3.0.0

24.5 AgsApplyBpm

AgsApplyBpm — apply bpm

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplyBpm *	ags_apply_bpm_new ()
#define	AGS_APPLY_BPM()
#define	AGS_APPLY_BPM_CLASS()
#define	AGS_APPLY_BPM_GET_CLASS()
#define	AGS_IS_APPLY_BPM()
#define	AGS_IS_APPLY_BPM_CLASS()
GType	ags_apply_bpm_get_type ()

Properties

double	bpm	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_BPM
struct	AgsApplyBpm
struct	AgsApplyBpmClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplyBpm
```

Includes

```
#include <ags/audio/task/recall/ags_apply_bpm.h>
```

Description

The AgsApplyBpm task applys bpm to AgsDelayAudio.

Functions

ags_apply_bpm_new ()

```
AgsApplyBpm~*
ags_apply_bpm_new (GObject *scope,
                   gdouble bpm);
```

Create a new instance of AgsApplyBpm.

Parameters

scope	the GObject	
bpm	the bpm to apply	

Returns

the new AgsApplyBpm

Since: 3.0.0

AGS_APPLY_BPM()

```
#define AGS_APPLY_BPM(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_BPM, AgsApplyBpm))
```

AGS_APPLY_BPM_CLASS()

```
#define AGS_APPLY_BPM_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_BPM, AgsApplyBpmClass))
```

AGS_APPLY_BPM_GET_CLASS()

```
#define AGS_APPLY_BPM_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_BPM, AgsApplyBpmClass))
```

AGS_IS_APPLY_BPM()

```
#define AGS_IS_APPLY_BPM(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_BPM))
```

AGS_IS_APPLY_BPM_CLASS()

```
#define AGS_IS_APPLY_BPM_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_BPM))
```

ags_apply_bpm_get_type ()

```
GType
ags_apply_bpm_get_type ();
```

Types and Values

AGS_TYPE_APPLY_BPM

```
#define AGS_TYPE_APPLY_BPM                 (ags_apply_bpm_get_type())
```

struct AgsApplyBpm

```
struct AgsApplyBpm;
```

struct AgsApplyBpmClass

```
struct AgsApplyBpmClass {
  AgsTaskClass task;
};
```

Property Details

The "bpm" property

```
  "bpm"                      double
```

The bpm to apply to scope.

Owner: AgsApplyBpm

Flags: Read / Write

Allowed values: >= 0

Default value: 120

Since: 3.0.0

The "scope" property

```
"scope"                    GObject~*
```

The assigned GObject as scope.

Owner: AgsApplyBpm

Flags: Read / Write

Since: 3.0.0

24.6 AgsApplyPresets

AgsApplyPresets — apply presets

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_apply_presets_soundcard ()
AgsApplyPresets *	ags_apply_presets_new ()
#define	AGS_APPLY_PRESETS()
#define	AGS_APPLY_PRESETS_CLASS()
#define	AGS_APPLY_PRESETS_GET_CLASS()
#define	AGS_IS_APPLY_PRESETS()
#define	AGS_IS_APPLY_PRESETS_CLASS()
GType	ags_apply_presets_get_type ()

Properties

guint	buffer-size	Read / Write
guint	format	Read / Write
guint	pcm-channels	Read / Write
guint	samplerate	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_PRESETS
struct	AgsApplyPresets
struct	AgsApplyPresetsClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsApplyPresets
```

Includes

```
#include <ags/audio/task/ags_apply_presets.h>
```

Description

The AgsApplyPresets task apply the specified presets.

Functions

ags_apply_presets_soundcard ()

```
void
ags_apply_presets_soundcard (AgsApplyPresets *apply_presets,
                             GObject *soundcard);
```

ags_apply_presets_new ()

```
AgsApplyPresets~*
ags_apply_presets_new (GObject *scope,
                       guint pcm_channels,
                       guint samplerate,
                       guint buffer_size,
                       guint format);
```

Creates a new instance of AgsApplyPresets.

Parameters

scope	a GObject	
pcm_channels	pcm channels	
samplerate	samplerate	
buffer_size	buffer size	
format	format	

Returns

the new AgsApplyPresets.

Since: 3.0.0

AGS_APPLY_PRESETS()

```
#define AGS_APPLY_PRESETS(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_PRESETS, AgsApplyPresets))
```

AGS_APPLY_PRESETS_CLASS()

```
#define AGS_APPLY_PRESETS_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_PRESETS, AgsApplyPresetsClass))
```

AGS_APPLY_PRESETS_GET_CLASS()

```
#define AGS_APPLY_PRESETS_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_PRESETS, AgsApplyPresetsClass))
```

AGS_IS_APPLY_PRESETS()

```
#define AGS_IS_APPLY_PRESETS(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_PRESETS))
```

AGS_IS_APPLY_PRESETS_CLASS()

```
#define AGS_IS_APPLY_PRESETS_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_PRESETS))
```

ags_apply_presets_get_type ()

```
GType
ags_apply_presets_get_type ();
```

Types and Values

AGS_TYPE_APPLY_PRESETS

```
#define AGS_TYPE_APPLY_PRESETS                 (ags_apply_presets_get_type())
```

struct AgsApplyPresets

```
struct AgsApplyPresets;
```

struct AgsApplyPresetsClass

```
struct AgsApplyPresetsClass {
  AgsTaskClass task;
};
```

Property Details

The "buffer-size" property

```
  "buffer-size"              guint
```

The count of buffer-size to apply.

Owner: AgsApplyPresets

Flags: Read / Write

Allowed values: <= 65535

Default value: 0

Since: 3.0.0

The "format" property

```
"format"                    guint
```

The count of format to apply.

Owner: AgsApplyPresets

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "pcm-channels" property

```
"pcm-channels"              guint
```

The count of pcm-channels to apply.

Owner: AgsApplyPresets

Flags: Read / Write

Allowed values: <= 65535

Default value: 0

Since: 3.0.0

The "samplerate" property

```
"samplerate"                guint
```

The count of samplerate to apply.

Owner: AgsApplyPresets

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "scope" property

```
"scope"                     GObject~*
```

The assigned GObject

Owner: AgsApplyPresets

Flags: Read / Write

Since: 3.0.0

24.7 AgsApplySequencerLength

AgsApplySequencerLength — apply sequencer length to delay audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySequencerLength *	ags_apply_sequencer_length_new ()
#define	AGS_APPLY_SEQUENCER_LENGTH()
#define	AGS_APPLY_SEQUENCER_LENGTH_CLASS()
#define	AGS_APPLY_SEQUENCER_LENGTH_GET_CLASS()
#define	AGS_IS_APPLY_SEQUENCER_LENGTH()
#define	AGS_IS_APPLY_SEQUENCER_LENGTH_CLASS()
GType	ags_apply_sequencer_length_get_type ()

Properties

GObject *	scope	Read / Write
double	sequencer-length	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SEQUENCER_LENGTH
struct	AgsApplySequencerLength
struct	AgsApplySequencerLengthClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplySequencerLength
```

Includes

```
#include <ags/audio/task/recall/ags_apply_sequencer_length.h>
```

Description

The AgsApplySequencerLength task applys sequencer length to AgsDelayAudio.

Functions

ags_apply_sequencer_length_new ()

```
AgsApplySequencerLength~*
ags_apply_sequencer_length_new (GObject *scope,
                                gdouble length);
```

Creates an AgsApplySequencerLength.

Parameters

scope	the GObject	
length	the sequencer length to apply	

Returns

an new AgsApplySequencerLength.

Since: 3.0.0

AGS_APPLY_SEQUENCER_LENGTH()

```
#define AGS_APPLY_SEQUENCER_LENGTH(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SEQUENCER_LENGTH, AgsApplySequencerLength))
```

AGS_APPLY_SEQUENCER_LENGTH_CLASS()

```
#define AGS_APPLY_SEQUENCER_LENGTH_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SEQUENCER_LENGTH, AgsApplySequencerLengthClass))
```

AGS_APPLY_SEQUENCER_LENGTH_GET_CLASS()

```
#define AGS_APPLY_SEQUENCER_LENGTH_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SEQUENCER_LENGTH, AgsApplySequencerLengthClass))
```

AGS_IS_APPLY_SEQUENCER_LENGTH()

```
#define AGS_IS_APPLY_SEQUENCER_LENGTH(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SEQUENCER_LENGTH))
```

AGS_IS_APPLY_SEQUENCER_LENGTH_CLASS()

```
#define AGS_IS_APPLY_SEQUENCER_LENGTH_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SEQUENCER_LENGTH))
```

ags_apply_sequencer_length_get_type ()

```
GType
ags_apply_sequencer_length_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SEQUENCER_LENGTH

```
#define AGS_TYPE_APPLY_SEQUENCER_LENGTH                 (ags_apply_sequencer_length_get_type ↩
    ())
```

struct AgsApplySequencerLength

```
struct AgsApplySequencerLength;
```

struct AgsApplySequencerLengthClass

```
struct AgsApplySequencerLengthClass {
  AgsTaskClass task;
};
```

Property Details

The "scope" property

```
  "scope"                     GObject~*
```

The assigned GObject as scope.

Owner: AgsApplySequencerLength

Flags: Read / Write

Since: 3.0.0

The "sequencer-length" property

```
  "sequencer-length"          double
```

The sequencer length to apply to scope.

Owner: AgsApplySequencerLength

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

24.8 AgsApplySoundConfig

AgsApplySoundConfig — apply sound config

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySoundConfig *	ags_apply_sound_config_new ()
#define	AGS_APPLY_SOUND_CONFIG()
#define	AGS_APPLY_SOUND_CONFIG_CLASS()
#define	AGS_APPLY_SOUND_CONFIG_GET_CLASS()
#define	AGS_IS_APPLY_SOUND_CONFIG()
#define	AGS_IS_APPLY_SOUND_CONFIG_CLASS()
GType	ags_apply_sound_config_get_type ()

Properties

char *	config-data	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SOUND_CONFIG
struct	AgsApplySoundConfig
struct	AgsApplySoundConfigClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplySoundConfig
```

Includes

```
#include <ags/audio/task/ags_apply_sound_config.h>
```

Description

The AgsApplySoundConfig task apply the specified sound config.

Functions

ags_apply_sound_config_new ()

```
AgsApplySoundConfig~*
ags_apply_sound_config_new (gchar *config_data);
```

Creates a new instance of AgsApplySoundConfig.

Parameters

config_data	the config file as string data

Returns

the new AgsApplySoundConfig.

Since: 3.0.0

AGS_APPLY_SOUND_CONFIG()

```
#define AGS_APPLY_SOUND_CONFIG(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SOUND_CONFIG, AgsApplySoundConfig))
```

AGS_APPLY_SOUND_CONFIG_CLASS()

```
#define AGS_APPLY_SOUND_CONFIG_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SOUND_CONFIG, AgsApplySoundConfigClass))
```

AGS_APPLY_SOUND_CONFIG_GET_CLASS()

```
#define AGS_APPLY_SOUND_CONFIG_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SOUND_CONFIG, AgsApplySoundConfigClass))
```

AGS_IS_APPLY_SOUND_CONFIG()

```
#define AGS_IS_APPLY_SOUND_CONFIG(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SOUND_CONFIG))
```

AGS_IS_APPLY_SOUND_CONFIG_CLASS()

```
#define AGS_IS_APPLY_SOUND_CONFIG_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SOUND_CONFIG))
```

ags_apply_sound_config_get_type ()

```
GType
ags_apply_sound_config_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SOUND_CONFIG

```
#define AGS_TYPE_APPLY_SOUND_CONFIG                 (ags_apply_sound_config_get_type())
```

struct AgsApplySoundConfig

```
struct AgsApplySoundConfig;
```

struct AgsApplySoundConfigClass

```
struct AgsApplySoundConfigClass {
  AgsTaskClass task;
};
```

Property Details

The "config-data" property

```
  "config-data"              char~*
```

The assigned sound config data as string.

Owner: AgsApplySoundConfig

Flags: Read / Write

Default value: NULL

Since: 3.0.0

24.9 AgsApplySynth

AgsApplySynth — apply synth to channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySynth *	ags_apply_synth_new ()
#define	AGS_APPLY_SYNTH()
#define	AGS_APPLY_SYNTH_CLASS()
#define	AGS_APPLY_SYNTH_GET_CLASS()
#define	AGS_IS_APPLY_SYNTH()
#define	AGS_IS_APPLY_SYNTH_CLASS()
GType	ags_apply_synth_get_type ()

Properties

double	base-note	Read / Write
guint	count	Read / Write
guint	requested-frame-count	Read / Write
AgsChannel *	start-channel	Read / Write
AgsSynthGenerator *	synth-generator	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SYNTH
struct	AgsApplySynth
struct	AgsApplySynthClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplySynth
```

Includes

```
#include <ags/audio/task/ags_apply_synth.h>
```

Description

The AgsApplySynth task apply the specified synth to channel.

Functions

ags_apply_synth_new ()

```
AgsApplySynth~*
ags_apply_synth_new (AgsSynthGenerator *synth_generator,
                     AgsChannel *start_channel,
                     gdouble base_note,
                     guint count);
```

Creates an AgsApplySynth.

Parameters

synth_generator	the AgsSynthGenerator	
start_channel	the start AgsChannel	
base_note	the base note	
count	the count of lines	

Returns

an new AgsApplySynth.

Since: 3.0.0

AGS_APPLY_SYNTH()

```
#define AGS_APPLY_SYNTH(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SYNTH, AgsApplySynth))
```

AGS_APPLY_SYNTH_CLASS()

```
#define AGS_APPLY_SYNTH_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SYNTH, AgsApplySynthClass))
```

AGS_APPLY_SYNTH_GET_CLASS()

```
#define AGS_APPLY_SYNTH_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SYNTH, AgsApplySynthClass))
```

AGS_IS_APPLY_SYNTH()

```
#define AGS_IS_APPLY_SYNTH(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SYNTH))
```

AGS_IS_APPLY_SYNTH_CLASS()

```
#define AGS_IS_APPLY_SYNTH_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SYNTH))
```

ags_apply_synth_get_type ()

```
GType
ags_apply_synth_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SYNTH

```
#define AGS_TYPE_APPLY_SYNTH                (ags_apply_synth_get_type())
```

struct AgsApplySynth

```
struct AgsApplySynth;
```

struct AgsApplySynthClass

```
struct AgsApplySynthClass {
  AgsTaskClass task;
};
```

Property Details

The "base-note" property

```
"base-note"                 double
```

The base-note to ramp up from.

Owner: AgsApplySynth

Flags: Read / Write

Allowed values: [-78,78]

Default value: -48

Since: 3.0.0

The "count" property

```
"count"                     guint
```

The count of channels to apply.

Owner: AgsApplySynth

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "requested-frame-count" property

```
"requested-frame-count"     guint
```

The frame count of audio signal to apply.

Owner: AgsApplySynth

Flags: Read / Write

Default value: 0

Since: 3.3.0

The "start-channel" property

```
"start-channel"             AgsChannel~*
```

The assigned AgsChannel

Owner: AgsApplySynth

Flags: Read / Write

Since: 3.0.0

The "synth-generator" property

```
"synth-generator"          AgsSynthGenerator~*
```

The assigned AgsSynthGenerator

Owner: AgsApplySynth

Flags: Read / Write

Since: 3.0.0

24.10 AgsApplySF2Synth

AgsApplySF2Synth — apply Soundfont2 synth to channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySF2Synth *	ags_apply_sf2_synth_new ()
#define	AGS_APPLY_SF2_SYNTH()
#define	AGS_APPLY_SF2_SYNTH_CLASS()
#define	AGS_APPLY_SF2_SYNTH_GET_CLASS()
#define	AGS_IS_APPLY_SF2_SYNTH()
#define	AGS_IS_APPLY_SF2_SYNTH_CLASS()
GType	ags_apply_sf2_synth_get_type ()

Properties

double	base-note	Read / Write
guint	count	Read / Write
guint	requested-frame-count	Read / Write
AgsSF2SynthGenerator *	sf2-synth-generator	Read / Write
AgsChannel *	start-channel	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SF2_SYNTH
struct	AgsApplySF2Synth
struct	AgsApplySF2SynthClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsApplySF2Synth
```

Includes

```
#include <ags/audio/task/ags_apply_sf2_synth.h>
```

Description

The AgsApplySF2Synth task apply the specified Soundfont2 synth to channel.

Functions

ags_apply_sf2_synth_new ()

```
AgsApplySF2Synth~*
ags_apply_sf2_synth_new (AgsSF2SynthGenerator *sf2_synth_generator,
                         AgsChannel *start_channel,
                         gdouble base_note,
                         guint count);
```

Creates an AgsApplySF2Synth.

Parameters

sf2_synth_generator	the AgsSF2SynthGenerator	
start_channel	the start AgsChannel	
base_note	the base note	
count	the count of lines	

Returns

an new AgsApplySF2Synth.

Since: 3.4.0

AGS_APPLY_SF2_SYNTH()

```
#define AGS_APPLY_SF2_SYNTH(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SF2_SYNTH, AgsApplySF2Synth))
```

AGS_APPLY_SF2_SYNTH_CLASS()

```
#define AGS_APPLY_SF2_SYNTH_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SF2_SYNTH, AgsApplySF2SynthClass))
```

AGS_APPLY_SF2_SYNTH_GET_CLASS()

```
#define AGS_APPLY_SF2_SYNTH_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SF2_SYNTH, AgsApplySF2SynthClass))
```

AGS_IS_APPLY_SF2_SYNTH()

```
#define AGS_IS_APPLY_SF2_SYNTH(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SF2_SYNTH))
```

AGS_IS_APPLY_SF2_SYNTH_CLASS()

```
#define AGS_IS_APPLY_SF2_SYNTH_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SF2_SYNTH))
```

ags_apply_sf2_synth_get_type ()

```
GType
ags_apply_sf2_synth_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SF2_SYNTH

```
#define AGS_TYPE_APPLY_SF2_SYNTH                   (ags_apply_sf2_synth_get_type())
```

struct AgsApplySF2Synth

```
struct AgsApplySF2Synth;
```

struct AgsApplySF2SynthClass

```
struct AgsApplySF2SynthClass {
  AgsTaskClass task;
};
```

Property Details

The "base-note" property

```
  "base-note"                 double
```

The base-note to ramp up from.

Owner: AgsApplySF2Synth

Flags: Read / Write

Allowed values: [-78,78]

Default value: -48

Since: 3.4.0

The "count" property

```
"count"                    guint
```

The count of channels to apply.

Owner: AgsApplySF2Synth

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "requested-frame-count" property

```
"requested-frame-count"    guint
```

The frame count of audio signal to apply.

Owner: AgsApplySF2Synth

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "sf2-synth-generator" property

```
"sf2-synth-generator"      AgsSF2SynthGenerator~*
```

The SF2 synth generator to apply.

Owner: AgsApplySF2Synth

Flags: Read / Write

The "start-channel" property

```
"start-channel"            AgsChannel~*
```

The assigned AgsChannel

Owner: AgsApplySF2Synth

Flags: Read / Write

Since: 3.4.0

24.11 AgsApplySF2MidiLocale

AgsApplySF2MidiLocale — apply Soundfont2 synth template

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySF2MidiLocale *	ags_apply_sf2_midi_locale_new ()
#define	AGS_APPLY_SF2_MIDI_LOCALE()
#define	AGS_APPLY_SF2_MIDI_LOCALE_CLASS()
#define	AGS_APPLY_SF2_MIDI_LOCALE_GET_CLASS()
#define	AGS_IS_APPLY_SF2_MIDI_LOCALE()
#define	AGS_IS_APPLY_SF2_MIDI_LOCALE_CLASS()
GType	ags_apply_sf2_midi_locale_get_type ()

Properties

gpointer	synth	Read / Write
gpointer	template	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SF2_MIDI_LOCALE
struct	AgsApplySF2MidiLocale
struct	AgsApplySF2MidiLocaleClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplySF2MidiLocale
```

Includes

```
#include <ags/audio/task/ags_apply_sf2_midi_locale.h>
```

Description

The AgsApplySF2MidiLocale task apply the specified Soundfont2 synth template.

Functions

ags_apply_sf2_midi_locale_new ()

```
AgsApplySF2MidiLocale~*
ags_apply_sf2_midi_locale_new (AgsSF2SynthUtil *template,
                               AgsSF2SynthUtil *synth);
```

Creates an AgsApplySF2MidiLocale.

Parameters

template	the template	
synth	the synth	

Returns

an new AgsApplySF2MidiLocale.

Since: 3.16.0

AGS_APPLY_SF2_MIDI_LOCALE()

```
#define AGS_APPLY_SF2_MIDI_LOCALE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SF2_MIDI_LOCALE, AgsApplySF2MidiLocale))
```

AGS_APPLY_SF2_MIDI_LOCALE_CLASS()

```
#define AGS_APPLY_SF2_MIDI_LOCALE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SF2_MIDI_LOCALE, AgsApplySF2MidiLocaleClass))
```

AGS_APPLY_SF2_MIDI_LOCALE_GET_CLASS()

```
#define AGS_APPLY_SF2_MIDI_LOCALE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SF2_MIDI_LOCALE, AgsApplySF2MidiLocaleClass))
```

AGS_IS_APPLY_SF2_MIDI_LOCALE()

```
#define AGS_IS_APPLY_SF2_MIDI_LOCALE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SF2_MIDI_LOCALE))
```

AGS_IS_APPLY_SF2_MIDI_LOCALE_CLASS()

```
#define AGS_IS_APPLY_SF2_MIDI_LOCALE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SF2_MIDI_LOCALE))
```

ags_apply_sf2_midi_locale_get_type ()

```
GType
ags_apply_sf2_midi_locale_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SF2_MIDI_LOCALE

```
#define AGS_TYPE_APPLY_SF2_MIDI_LOCALE                (ags_apply_sf2_midi_locale_get_type() ↩
    )
```

struct AgsApplySF2MidiLocale

```
struct AgsApplySF2MidiLocale;
```

struct AgsApplySF2MidiLocaleClass

```
struct AgsApplySF2MidiLocaleClass {
  AgsTaskClass task;
};
```

Property Details

The "synth" property

```
  "synth"                    gpointer
```

The synth to apply.

Owner: AgsApplySF2MidiLocale

Flags: Read / Write

Since: 3.16.0

The "template" property

```
  "template"                 gpointer
```

The template to apply.

Owner: AgsApplySF2MidiLocale

Flags: Read / Write

Since: 3.16.0

24.12 AgsApplySFZSynth

AgsApplySFZSynth — apply SFZ synth to channel

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySFZSynth *	ags_apply_sfz_synth_new ()
#define	AGS_APPLY_SFZ_SYNTH()
#define	AGS_APPLY_SFZ_SYNTH_CLASS()
#define	AGS_APPLY_SFZ_SYNTH_GET_CLASS()
#define	AGS_IS_APPLY_SFZ_SYNTH()

#define	AGS_IS_APPLY_SFZ_SYNTH_CLASS()
GType	ags_apply_sfz_synth_get_type ()

Properties

double	base-note	Read / Write
guint	count	Read / Write
guint	requested-frame-count	Read / Write
AgsSFZSynthGenerator *	sfz-synth-generator	Read / Write
AgsChannel *	start-channel	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SFZ_SYNTH
struct	AgsApplySFZSynth
struct	AgsApplySFZSynthClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplySFZSynth
```

Includes

```
#include <ags/audio/task/ags_apply_sfz_synth.h>
```

Description

The AgsApplySFZSynth task apply the specified SFZ synth to channel.

Functions

ags_apply_sfz_synth_new ()

```
AgsApplySFZSynth~*
ags_apply_sfz_synth_new (AgsSFZSynthGenerator *sfz_synth_generator,
                         AgsChannel *start_channel,
                         gdouble base_note,
                         guint count);
```

Creates an AgsApplySFZSynth.

Parameters

sfz_synth_generator	the AgsSFZSynthGenerator	
start_channel	the start AgsChannel	
base_note	the base note	
count	the count of lines	

Returns

an new AgsApplySFZSynth.

Since: 3.4.0

AGS_APPLY_SFZ_SYNTH()

```
#define AGS_APPLY_SFZ_SYNTH(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SFZ_SYNTH, AgsApplySFZSynth))
```

AGS_APPLY_SFZ_SYNTH_CLASS()

```
#define AGS_APPLY_SFZ_SYNTH_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SFZ_SYNTH, AgsApplySFZSynthClass))
```

AGS_APPLY_SFZ_SYNTH_GET_CLASS()

```
#define AGS_APPLY_SFZ_SYNTH_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SFZ_SYNTH, AgsApplySFZSynthClass))
```

AGS_IS_APPLY_SFZ_SYNTH()

```
#define AGS_IS_APPLY_SFZ_SYNTH(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SFZ_SYNTH))
```

AGS_IS_APPLY_SFZ_SYNTH_CLASS()

```
#define AGS_IS_APPLY_SFZ_SYNTH_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SFZ_SYNTH))
```

ags_apply_sfz_synth_get_type ()

```
GType
ags_apply_sfz_synth_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SFZ_SYNTH

```
#define AGS_TYPE_APPLY_SFZ_SYNTH                (ags_apply_sfz_synth_get_type())
```

struct AgsApplySFZSynth

```
struct AgsApplySFZSynth;
```

struct AgsApplySFZSynthClass

```
struct AgsApplySFZSynthClass {
  AgsTaskClass task;
};
```

Property Details

The "base-note" property

```
  "base-note"                 double
```

The base-note to ramp up from.

Owner: AgsApplySFZSynth

Flags: Read / Write

Allowed values: [-78,78]

Default value: -48

Since: 3.4.0

The "count" property

```
  "count"                     guint
```

The count of channels to apply.

Owner: AgsApplySFZSynth

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "requested-frame-count" property

```
  "requested-frame-count"     guint
```

The frame count of audio signal to apply.

Owner: AgsApplySFZSynth

Flags: Read / Write

Default value: 0

Since: 3.4.0

The "sfz-synth-generator" property

```
  "sfz-synth-generator"       AgsSFZSynthGenerator~*
```

The SFZ synth generator to apply.

Owner: AgsApplySFZSynth

Flags: Read / Write

The "start-channel" property

```
"start-channel"            AgsChannel~*
```

The assigned AgsChannel

Owner: AgsApplySFZSynth

Flags: Read / Write

Since: 3.4.0

24.13 AgsApplySFZInstrument

AgsApplySFZInstrument — apply SFZ synth template

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplySFZInstrument *	ags_apply_sfz_instrument_new ()
#define	AGS_APPLY_SFZ_INSTRUMENT()
#define	AGS_APPLY_SFZ_INSTRUMENT_CLASS()
#define	AGS_APPLY_SFZ_INSTRUMENT_GET_CLASS()
#define	AGS_IS_APPLY_SFZ_INSTRUMENT()
#define	AGS_IS_APPLY_SFZ_INSTRUMENT_CLASS()
GType	ags_apply_sfz_instrument_get_type ()

Properties

gpointer	synth	Read / Write
gpointer	template	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_SFZ_INSTRUMENT
struct	AgsApplySFZInstrument
struct	AgsApplySFZInstrumentClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsApplySFZInstrument
```

Includes

```
#include <ags/audio/task/ags_apply_sfz_instrument.h>
```

Description

The AgsApplySFZInstrument task apply the specified SFZ synth template.

Functions

ags_apply_sfz_instrument_new ()

```
AgsApplySFZInstrument~*
ags_apply_sfz_instrument_new (AgsSFZSynthUtil *template,
                              AgsSFZSynthUtil *synth);
```

Creates an AgsApplySFZInstrument.

Parameters

template	the template	
synth	the synth	

Returns

an new AgsApplySFZInstrument.

Since: 3.17.0

AGS_APPLY_SFZ_INSTRUMENT()

```
#define AGS_APPLY_SFZ_INSTRUMENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_SFZ_INSTRUMENT, AgsApplySFZInstrument))
```

AGS_APPLY_SFZ_INSTRUMENT_CLASS()

```
#define AGS_APPLY_SFZ_INSTRUMENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_SFZ_INSTRUMENT, AgsApplySFZInstrumentClass))
```

AGS_APPLY_SFZ_INSTRUMENT_GET_CLASS()

```
#define AGS_APPLY_SFZ_INSTRUMENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_SFZ_INSTRUMENT, AgsApplySFZInstrumentClass))
```

AGS_IS_APPLY_SFZ_INSTRUMENT()

```
#define AGS_IS_APPLY_SFZ_INSTRUMENT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_SFZ_INSTRUMENT))
```

AGS_IS_APPLY_SFZ_INSTRUMENT_CLASS()

```
#define AGS_IS_APPLY_SFZ_INSTRUMENT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_SFZ_INSTRUMENT))
```

ags_apply_sfz_instrument_get_type ()

```
GType
ags_apply_sfz_instrument_get_type ();
```

Types and Values

AGS_TYPE_APPLY_SFZ_INSTRUMENT

```
#define AGS_TYPE_APPLY_SFZ_INSTRUMENT                (ags_apply_sfz_instrument_get_type())
```

struct AgsApplySFZInstrument

```
struct AgsApplySFZInstrument;
```

struct AgsApplySFZInstrumentClass

```
struct AgsApplySFZInstrumentClass {
  AgsTaskClass task;
};
```

Property Details

The "synth" property

```
  "synth"                    gpointer
```

The synth to apply.

Owner: AgsApplySFZInstrument

Flags: Read / Write

Since: 3.17.0

The "template" property

```
  "template"                 gpointer
```

The template to apply.

Owner: AgsApplySFZInstrument

Flags: Read / Write

Since: 3.17.0

24.14 AgsApplyTact

AgsApplyTact — apply tact to delay audio

Stability Level

Stable, unless otherwise indicated

Functions

AgsApplyTact *	ags_apply_tact_new ()
#define	AGS_APPLY_TACT()
#define	AGS_APPLY_TACT_CLASS()
#define	AGS_APPLY_TACT_GET_CLASS()
#define	AGS_IS_APPLY_TACT()
#define	AGS_IS_APPLY_TACT_CLASS()
GType	ags_apply_tact_get_type ()

Properties

GObject *	scope	Read / Write
double	tact	Read / Write

Types and Values

#define	AGS_TYPE_APPLY_TACT
struct	AgsApplyTact
struct	AgsApplyTactClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsApplyTact
```

Includes

```
#include <ags/audio/task/recall/ags_apply_tact.h>
```

Description

The AgsApplyTact task applys tact to AgsTactable.

Functions

ags_apply_tact_new ()

```
AgsApplyTact~*
ags_apply_tact_new (GObject *scope,
                    gdouble tact);
```

Creates an AgsApplyTact.

Parameters

scope	the GObject	
tact	the tact to apply	

Returns

an new AgsApplyTact.

Since: 3.0.0

AGS_APPLY_TACT()

```
#define AGS_APPLY_TACT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLY_TACT, AgsApplyTact))
```

AGS_APPLY_TACT_CLASS()

```
#define AGS_APPLY_TACT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_APPLY_TACT, AgsApplyTactClass))
```

AGS_APPLY_TACT_GET_CLASS()

```
#define AGS_APPLY_TACT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_APPLY_TACT, AgsApplyTactClass))
```

AGS_IS_APPLY_TACT()

```
#define AGS_IS_APPLY_TACT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLY_TACT))
```

AGS_IS_APPLY_TACT_CLASS()

```
#define AGS_IS_APPLY_TACT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_APPLY_TACT))
```

ags_apply_tact_get_type ()

```
GType
ags_apply_tact_get_type ();
```

Types and Values

AGS_TYPE_APPLY_TACT

```
#define AGS_TYPE_APPLY_TACT                 (ags_apply_tact_get_type())
```

struct AgsApplyTact

```
struct AgsApplyTact;
```

struct AgsApplyTactClass

```
struct AgsApplyTactClass {
  AgsTaskClass task;
};
```

Property Details

The "scope" property

```
  "scope"                    GObject~*
```

The assigned GObject as scope.

Owner: AgsApplyTact

Flags: Read / Write

Since: 3.0.0

The "tact" property

```
  "tact"                     double
```

The tact to apply to scope.

Owner: AgsApplyTact

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

24.15 AgsCancelAudio

AgsCancelAudio — cancel audio task

Stability Level

Stable, unless otherwise indicated

Functions

AgsCancelAudio *	ags_cancel_audio_new ()
#define	AGS_CANCEL_AUDIO()
#define	AGS_CANCEL_AUDIO_CLASS()
#define	AGS_CANCEL_AUDIO_GET_CLASS()
#define	AGS_IS_CANCEL_AUDIO()
#define	AGS_IS_CANCEL_AUDIO_CLASS()
GType	ags_cancel_audio_get_type ()

Properties

AgsAudio *	audio	Read / Write
int	sound-scope	Read / Write

Types and Values

#define	AGS_TYPE_CANCEL_AUDIO
struct	AgsCancelAudio
struct	AgsCancelAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsCancelAudio
```

Includes

```
#include <ags/audio/task/ags_cancel_audio.h>
```

Description

The AgsCancelAudio task cancels AgsAudio playback.

Functions

ags_cancel_audio_new ()

```
AgsCancelAudio~*
ags_cancel_audio_new (AgsAudio *audio,
                      gint sound_scope);
```

Create a new instance of AgsCancelAudio.

Parameters

audio	the AgsAudio to cancel	
sound_scope	the AgsSoundScope or -1 for all	

Returns

the new AgsCancelAudio.

Since: 3.0.0

AGS_CANCEL_AUDIO()

```
#define AGS_CANCEL_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CANCEL_AUDIO, AgsCancelAudio))
```

AGS_CANCEL_AUDIO_CLASS()

```
#define AGS_CANCEL_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CANCEL_AUDIO, AgsCancelAudioClass))
```

AGS_CANCEL_AUDIO_GET_CLASS()

```
#define AGS_CANCEL_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_CANCEL_AUDIO, AgsCancelAudioClass))
```

AGS_IS_CANCEL_AUDIO()

```
#define AGS_IS_CANCEL_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CANCEL_AUDIO))
```

AGS_IS_CANCEL_AUDIO_CLASS()

```
#define AGS_IS_CANCEL_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_CANCEL_AUDIO))
```

ags_cancel_audio_get_type ()

```
GType
ags_cancel_audio_get_type ();
```

Types and Values

AGS_TYPE_CANCEL_AUDIO

```
#define AGS_TYPE_CANCEL_AUDIO                (ags_cancel_audio_get_type())
```

struct AgsCancelAudio

```
struct AgsCancelAudio;
```

struct AgsCancelAudioClass

```
struct AgsCancelAudioClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsCancelAudio

Flags: Read / Write

Since: 3.0.0

The "sound-scope" property

```
  "sound-scope"               int
```

The effects sound-scope.

Owner: AgsCancelAudio

Flags: Read / Write

Allowed values: [-1,5]

Default value: -1

Since: 3.0.0

24.16 AgsCancelChannel

AgsCancelChannel — cancel channel task

Stability Level

Stable, unless otherwise indicated

Functions

AgsCancelChannel *	ags_cancel_channel_new ()
#define	AGS_CANCEL_CHANNEL()
#define	AGS_CANCEL_CHANNEL_CLASS()
#define	AGS_CANCEL_CHANNEL_GET_CLASS()
#define	AGS_IS_CANCEL_CHANNEL()
#define	AGS_IS_CANCEL_CHANNEL_CLASS()
GType	ags_cancel_channel_get_type ()

Properties

AgsChannel *	channel	Read / Write
int	sound-scope	Read / Write

Types and Values

#define	AGS_TYPE_CANCEL_CHANNEL
struct	AgsCancelChannel
struct	AgsCancelChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsCancelChannel
```

Includes

```
#include <ags/audio/task/ags_cancel_channel.h>
```

Description

The AgsCancelChannel task cancels AgsChannel playback.

Functions

ags_cancel_channel_new ()

```
AgsCancelChannel~*
ags_cancel_channel_new (AgsChannel *channel,
                        gint sound_scope);
```

Create a new instance of AgsCancelChannel.

Parameters

channel	the AgsChannel to cancel	
sound_scope	the AgsSoundScope or -1 for all	

Returns

the new AgsCancelChannel.

Since: 3.0.0

AGS_CANCEL_CHANNEL()

```
#define AGS_CANCEL_CHANNEL(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CANCEL_CHANNEL, AgsCancelChannel))
```

AGS_CANCEL_CHANNEL_CLASS()

```
#define AGS_CANCEL_CHANNEL_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CANCEL_CHANNEL, AgsCancelChannelClass))
```

AGS_CANCEL_CHANNEL_GET_CLASS()

```
#define AGS_CANCEL_CHANNEL_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_CANCEL_CHANNEL, AgsCancelChannelClass))
```

AGS_IS_CANCEL_CHANNEL()

```
#define AGS_IS_CANCEL_CHANNEL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CANCEL_CHANNEL))
```

AGS_IS_CANCEL_CHANNEL_CLASS()

```
#define AGS_IS_CANCEL_CHANNEL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_CANCEL_CHANNEL))
```

ags_cancel_channel_get_type ()

```
GType
ags_cancel_channel_get_type ();
```

Types and Values

AGS_TYPE_CANCEL_CHANNEL

```
#define AGS_TYPE_CANCEL_CHANNEL                 (ags_cancel_channel_get_type())
```

struct AgsCancelChannel

```
struct AgsCancelChannel;
```

struct AgsCancelChannelClass

```
struct AgsCancelChannelClass {
  AgsTaskClass task;
};
```

Property Details

The "channel" property

```
  "channel"                   AgsChannel~*
```

The assigned AgsChannel

Owner: AgsCancelChannel

Flags: Read / Write

Since: 3.0.0

The "sound-scope" property

```
  "sound-scope"               int
```

The effects sound-scope.

Owner: AgsCancelChannel

Flags: Read / Write

Allowed values: [-1,5]

Default value: -1

Since: 3.0.0

24.17 AgsClearAudioSignal

AgsClearAudioSignal — clear audio signal object

Stability Level

Stable, unless otherwise indicated

Functions

AgsClearAudioSignal *	ags_clear_audio_signal_new ()
#define	AGS_CLEAR_AUDIO_SIGNAL()
#define	AGS_CLEAR_AUDIO_SIGNAL_CLASS()
#define	AGS_CLEAR_AUDIO_SIGNAL_GET_CLASS()
#define	AGS_IS_CLEAR_AUDIO_SIGNAL()
#define	AGS_IS_CLEAR_AUDIO_SIGNAL_CLASS()
GType	ags_clear_audio_signal_get_type ()

Properties

AgsAudioSignal *	audio-signal	Read / Write

Types and Values

#define	AGS_TYPE_CLEAR_AUDIO_SIGNAL
struct	AgsClearAudioSignal
struct	AgsClearAudioSignalClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsClearAudioSignal
```

Includes

```
#include <ags/audio/task/ags_clear_audio_signal.h>
```

Description

The AgsClearAudioSignal task clears AgsAudioSignal.

Functions

ags_clear_audio_signal_new ()

```
AgsClearAudioSignal~*
ags_clear_audio_signal_new (AgsAudioSignal *audio_signal);
```

Create a new instance of AgsClearAudioSignal.

Parameters

audio_signal	the AgsAudioSignal to clear	

Returns

the new AgsClearAudioSignal.

Since: 3.0.0

AGS_CLEAR_AUDIO_SIGNAL()

```
#define AGS_CLEAR_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CLEAR_AUDIO_SIGNAL, AgsClearAudioSignal))
```

AGS_CLEAR_AUDIO_SIGNAL_CLASS()

```
#define AGS_CLEAR_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CLEAR_AUDIO_SIGNAL, AgsClearAudioSignalClass))
```

AGS_CLEAR_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_CLEAR_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_CLEAR_AUDIO_SIGNAL, AgsClearAudioSignalClass))
```

AGS_IS_CLEAR_AUDIO_SIGNAL()

```
#define AGS_IS_CLEAR_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CLEAR_AUDIO_SIGNAL))
```

AGS_IS_CLEAR_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_CLEAR_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_CLEAR_AUDIO_SIGNAL))
```

ags_clear_audio_signal_get_type ()

```
GType
ags_clear_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_CLEAR_AUDIO_SIGNAL

```
#define AGS_TYPE_CLEAR_AUDIO_SIGNAL                (ags_clear_audio_signal_get_type())
```

struct AgsClearAudioSignal

```
struct AgsClearAudioSignal;
```

struct AgsClearAudioSignalClass

```
struct AgsClearAudioSignalClass {
  AgsTaskClass task;
};
```

Property Details

The "audio-signal" property

```
  "audio-signal"              AgsAudioSignal~*
```

The assigned AgsAudioSignal

Owner: AgsClearAudioSignal

Flags: Read / Write

Since: 3.0.0

24.18 AgsClearBuffer

AgsClearBuffer — switch buffer flag of device

Stability Level

Stable, unless otherwise indicated

Functions

AgsClearBuffer *	ags_clear_buffer_new ()
#define	AGS_CLEAR_BUFFER()
#define	AGS_CLEAR_BUFFER_CLASS()
#define	AGS_CLEAR_BUFFER_GET_CLASS()
#define	AGS_IS_CLEAR_BUFFER()
#define	AGS_IS_CLEAR_BUFFER_CLASS()
GType	ags_clear_buffer_get_type ()

Properties

GObject *	device	Read / Write

Types and Values

#define	AGS_TYPE_CLEAR_BUFFER
struct	AgsClearBuffer
struct	AgsClearBufferClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsClearBuffer
```

Includes

```
#include <ags/audio/task/ags_clear_buffer.h>
```

Description

The AgsClearBuffer task switches the buffer flag of device.

Functions

ags_clear_buffer_new ()

```
AgsClearBuffer~*
ags_clear_buffer_new (GObject *device);
```

Create a new instance of AgsClearBuffer.

Parameters

device	the AgsSoundcard or AgsSequencer

Returns

the new AgsClearBuffer.

Since: 3.0.0

AGS_CLEAR_BUFFER()

```
#define AGS_CLEAR_BUFFER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CLEAR_BUFFER, AgsClearBuffer))
```

AGS_CLEAR_BUFFER_CLASS()

```
#define AGS_CLEAR_BUFFER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CLEAR_BUFFER, AgsClearBufferClass))
```

AGS_CLEAR_BUFFER_GET_CLASS()

```
#define AGS_CLEAR_BUFFER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_CLEAR_BUFFER, AgsClearBufferClass))
```

AGS_IS_CLEAR_BUFFER()

```
#define AGS_IS_CLEAR_BUFFER(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CLEAR_BUFFER))
```

AGS_IS_CLEAR_BUFFER_CLASS()

```
#define AGS_IS_CLEAR_BUFFER_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_CLEAR_BUFFER))
```

ags_clear_buffer_get_type ()

```
GType
ags_clear_buffer_get_type ();
```

Types and Values

AGS_TYPE_CLEAR_BUFFER

```
#define AGS_TYPE_CLEAR_BUFFER                   (ags_clear_buffer_get_type())
```

struct AgsClearBuffer

```
struct AgsClearBuffer;
```

struct AgsClearBufferClass

```
struct AgsClearBufferClass {
  AgsTaskClass task;
};
```

Property Details

The "device" property

```
  "device"                    GObject~*
```

The assigned AgsSoundcard or AgsSequencer

Owner: AgsClearBuffer

Flags: Read / Write

Since: 3.0.0

24.19 AgsCropNote

AgsCropNote — crop notation

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_crop_note_set_selection ()
AgsCropNote *	ags_crop_note_new ()
#define	AGS_CROP_NOTE()
#define	AGS_CROP_NOTE_CLASS()
#define	AGS_CROP_NOTE_GET_CLASS()
#define	AGS_IS_CROP_NOTE()
#define	AGS_IS_CROP_NOTE_CLASS()
GType	ags_crop_note_get_type ()

Properties

gboolean	absolute	Read / Write
AgsAudio *	audio	Read / Write
gboolean	do-resize	Read / Write
gboolean	in-place	Read / Write
AgsNotation *	notation	Read / Write
gpointer	selection	Read / Write
int	x-crop	Read / Write
guint	x-padding	Read / Write

Types and Values

#define	AGS_CROP_NOTE_DEFAULT_X_LENGTH
#define	AGS_TYPE_CROP_NOTE
struct	AgsCropNote
struct	AgsCropNoteClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsCropNote
```

Includes

```
#include <ags/audio/task/ags_crop_note.h>
```

Description

The AgsCropNote task crops AgsNotation.

Functions

ags_crop_note_set_selection ()

```
void
ags_crop_note_set_selection (AgsCropNote *crop_note,
                             GList *selection);
```

Set *selection* of *crop_note*.

Parameters

crop_note	the AgsCropNote	
selection	the selection as GList.	*[element-type AgsAudio.Note][transfer none]*

Since: 3.2.3

ags_crop_note_new ()

```
AgsCropNote~*
ags_crop_note_new (AgsAudio *audio,
                   AgsNotation *notation,
                   GList *selection,
                   guint x_padding,
                   gint x_crop,
                   gboolean absolute,
                   gboolean in_place,
                   gboolean do_resize);
```

WARNING you need to provide AgsAudio as a property. Creates an AgsCropNote task. Note either *in_place* or *do_resize* shall be TRUE else it won't have any effect.

Parameters

audio	the AgsAudio	
notation	the AgsNotation	
selection	the selection as GList.	*[element-type AgsAudio.Note][transfer none]*
x_padding	the x padding to use	
x_crop	the amout to crop	
absolute	if TRUE from absolute position, otherwise relative	
in_place	if TRUE crop in place, otherwise grow relative offset	
do_resize	if TRUE resize notation, otherwise not	

Returns

a new AgsCropNote

Since: 3.0.0

AGS_CROP_NOTE()

```
#define AGS_CROP_NOTE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CROP_NOTE, AgsCropNote))
```

AGS_CROP_NOTE_CLASS()

```
#define AGS_CROP_NOTE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CROP_NOTE, AgsCropNoteClass))
```

AGS_CROP_NOTE_GET_CLASS()

```
#define AGS_CROP_NOTE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_CROP_NOTE, AgsCropNoteClass))
```

AGS_IS_CROP_NOTE()

```
#define AGS_IS_CROP_NOTE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CROP_NOTE))
```

AGS_IS_CROP_NOTE_CLASS()

```
#define AGS_IS_CROP_NOTE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_CROP_NOTE))
```

ags_crop_note_get_type ()

```
GType
ags_crop_note_get_type ();
```

Types and Values

AGS_CROP_NOTE_DEFAULT_X_LENGTH

```
#define AGS_CROP_NOTE_DEFAULT_X_LENGTH (16 * 16 * 1200)
```

AGS_TYPE_CROP_NOTE

```
#define AGS_TYPE_CROP_NOTE                (ags_crop_note_get_type())
```

struct AgsCropNote

```
struct AgsCropNote;
```

struct AgsCropNoteClass

```
struct AgsCropNoteClass {
  AgsTaskClass task;
};
```

Property Details

The "absolute" property

```
  "absolute"                  gboolean
```

Crop notation by absolute position.

Owner: AgsCropNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsCropNote

Flags: Read / Write

Since: 3.0.0

The "do-resize" property

```
  "do-resize"                 gboolean
```

Crop notation do resize.

Owner: AgsCropNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "in-place" property

```
  "in-place"                  gboolean
```

Crop notation in place.

Owner: AgsCropNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "notation" property

```
"notation"                  AgsNotation~*
```

The assigned AgsNotation

Owner: AgsCropNote

Flags: Read / Write

Since: 3.0.0

The "selection" property

```
"selection"                 gpointer
```

The assigned AgsNote

[transfer full]

Owner: AgsCropNote

Flags: Read / Write

Since: 3.0.0

The "x-crop" property

```
"x-crop"                    int
```

Crop notation by x-crop amount.

Owner: AgsCropNote

Flags: Read / Write

Allowed values: [-307200,307200]

Default value: 0

Since: 3.0.0

The "x-padding" property

```
"x-padding"                 guint
```

Crop notation with x padding.

Owner: AgsCropNote

Flags: Read / Write

Allowed values: <= 307200

Default value: 0

Since: 3.0.0

24.20 AgsExportOutput

AgsExportOutput — export output task

Stability Level

Stable, unless otherwise indicated

Functions

AgsExportOutput *	ags_export_output_new ()
#define	AGS_EXPORT_OUTPUT()
#define	AGS_EXPORT_OUTPUT_CLASS()
#define	AGS_EXPORT_OUTPUT_GET_CLASS()
#define	AGS_IS_EXPORT_OUTPUT()
#define	AGS_IS_EXPORT_OUTPUT_CLASS()
GType	ags_export_output_get_type ()

Properties

AgsExportThread *	export-thread	Read / Write
char *	filename	Read / Write
guint	format	Read / Write
gboolean	live-performance	Read / Write
GObject *	soundcard	Read / Write
guint	tic	Read / Write

Types and Values

enum	AgsExportOutputFormat
#define	AGS_TYPE_EXPORT_OUTPUT
struct	AgsExportOutput
struct	AgsExportOutputClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsExportOutput
```

Includes

```
#include <ags/audio/task/ags_export_output.h>
```

Description

The AgsExportOutput task exports GObject to file.

Functions

ags_export_output_new ()

```
AgsExportOutput~*
ags_export_output_new (AgsExportThread *export_thread,
                       GObject *soundcard,
                       gchar *filename,
                       guint tic,
                       gboolean live_performance);
```

Creates an AgsExportOutput.

Parameters

export_thread	the AgsExportThread to start	
soundcard	the GObject to export	
filename	the filename to save	
tic	stream duration in tact	
live_performance	if TRUE export is done during real-time	

Returns

an new AgsExportOutput.

Since: 3.0.0

AGS_EXPORT_OUTPUT()

```
#define AGS_EXPORT_OUTPUT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_EXPORT_OUTPUT, AgsExportOutput))
```

AGS_EXPORT_OUTPUT_CLASS()

```
#define AGS_EXPORT_OUTPUT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_EXPORT_OUTPUT, AgsExportOutputClass))
```

AGS_EXPORT_OUTPUT_GET_CLASS()

```
#define AGS_EXPORT_OUTPUT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_EXPORT_OUTPUT, AgsExportOutputClass))
```

AGS_IS_EXPORT_OUTPUT()

```
#define AGS_IS_EXPORT_OUTPUT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_EXPORT_OUTPUT))
```

AGS_IS_EXPORT_OUTPUT_CLASS()

```
#define AGS_IS_EXPORT_OUTPUT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_EXPORT_OUTPUT))
```

ags_export_output_get_type ()

```
GType
ags_export_output_get_type ();
```

Types and Values

enum AgsExportOutputFormat

Enum values to control output format of AgsExportOutput.

Members

AGS_EXPORT_OUTPUT_FORMAT_WAV	use WAV format
AGS_EXPORT_OUTPUT_FORMAT_OGG	use OGG format
AGS_EXPORT_OUTPUT_FORMAT_FLAC	use FLAC format
AGS_EXPORT_OUTPUT_FORMAT_AIFF	use AIFF format

AGS_TYPE_EXPORT_OUTPUT

```
#define AGS_TYPE_EXPORT_OUTPUT                 (ags_export_output_get_type())
```

struct AgsExportOutput

```
struct AgsExportOutput;
```

struct AgsExportOutputClass

```
struct AgsExportOutputClass {
  AgsTaskClass task;
};
```

Property Details

The "export-thread" property

```
"export-thread"            AgsExportThread~*
```

The assigned AgsExportThread

Owner: AgsExportOutput

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
"filename"                 char~*
```

The assigned filename.

Owner: AgsExportOutput

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "format" property

```
"format"                   guint
```

Format to use.

Owner: AgsExportOutput

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "live-performance" property

```
"live-performance"         gboolean
```

Do output the audio export live.

Owner: AgsExportOutput

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "soundcard" property

```
"soundcard"                GObject~*
```

The assigned AgsSoundcard

Owner: AgsExportOutput

Flags: Read / Write

Since: 3.0.0

The "tic" property

```
"tic"                    guint
```

Tic offset of output as end tic of it.

Owner: AgsExportOutput

Flags: Read / Write

Default value: 0

Since: 3.0.0

24.21 AgsFreeSelection

AgsFreeSelection — free selection object

Stability Level

Stable, unless otherwise indicated

Functions

AgsFreeSelection *	ags_free_selection_new ()
#define	AGS_FREE_SELECTION()
#define	AGS_FREE_SELECTION_CLASS()
#define	AGS_FREE_SELECTION_GET_CLASS()
#define	AGS_IS_FREE_SELECTION()
#define	AGS_IS_FREE_SELECTION_CLASS()
GType	ags_free_selection_get_type ()

Properties

AgsNotation *	notation	Read / Write

Types and Values

#define	AGS_TYPE_FREE_SELECTION
struct	AgsFreeSelection
struct	AgsFreeSelectionClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsFreeSelection
```

Includes

```
#include <ags/audio/task/ags_free_selection.h>
```

Description

The AgsFreeSelection task frees selection of AgsNotation.

Functions

ags_free_selection_new ()

```
AgsFreeSelection~*
ags_free_selection_new (AgsNotation *notation);
```

Create a new instance of AgsFreeSelection.

Parameters

notation	the AgsNotation

Returns

the new AgsFreeSelection.

Since: 3.0.0

AGS_FREE_SELECTION()

```
#define AGS_FREE_SELECTION(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FREE_SELECTION, AgsFreeSelection))
```

AGS_FREE_SELECTION_CLASS()

```
#define AGS_FREE_SELECTION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FREE_SELECTION, AgsFreeSelectionClass))
```

AGS_FREE_SELECTION_GET_CLASS()

```
#define AGS_FREE_SELECTION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_FREE_SELECTION, AgsFreeSelectionClass))
```

AGS_IS_FREE_SELECTION()

```
#define AGS_IS_FREE_SELECTION(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_FREE_SELECTION))
```

AGS_IS_FREE_SELECTION_CLASS()

```
#define AGS_IS_FREE_SELECTION_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_FREE_SELECTION))
```

ags_free_selection_get_type ()

```
GType
ags_free_selection_get_type ();
```

Types and Values

AGS_TYPE_FREE_SELECTION

```
#define AGS_TYPE_FREE_SELECTION                (ags_free_selection_get_type())
```

struct AgsFreeSelection

```
struct AgsFreeSelection;
```

struct AgsFreeSelectionClass

```
struct AgsFreeSelectionClass {
  AgsTaskClass task;
};
```

Property Details

The "notation" property

```
  "notation"                 AgsNotation~*
```

The assigned AgsNotation

Owner: AgsFreeSelection

Flags: Read / Write

Since: 3.0.0

24.22 AgsLinkChannel

AgsLinkChannel — link channel task

Stability Level

Stable, unless otherwise indicated

Functions

AgsLinkChannel *	ags_link_channel_new ()
#define	AGS_IS_LINK_CHANNEL()
#define	AGS_IS_LINK_CHANNEL_CLASS()
#define	AGS_LINK_CHANNEL()
#define	AGS_LINK_CHANNEL_CLASS()
#define	AGS_LINK_CHANNEL_GET_CLASS()
GType	ags_link_channel_get_type ()

Properties

AgsChannel *	channel	Read / Write
gpointer	error	Read / Write
AgsChannel *	link	Read / Write

Types and Values

#define	AGS_TYPE_LINK_CHANNEL
struct	AgsLinkChannel
struct	AgsLinkChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsLinkChannel
```

Includes

```
#include <ags/audio/task/ags_link_channel.h>
```

Description

The AgsLinkChannel task links AgsChannel.

Functions

ags_link_channel_new ()

```
AgsLinkChannel~*
ags_link_channel_new (AgsChannel *channel,
                      AgsChannel *link);
```

Create a new instance of AgsLinkChannel.

Parameters

channel	the AgsChannel	
link	the AgsChannel to be linked, may be NULL	

Returns

the new AgsLinkChannel.

Since: 3.0.0

AGS_IS_LINK_CHANNEL()

```
#define AGS_IS_LINK_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_LINK_CHANNEL))
```

AGS_IS_LINK_CHANNEL_CLASS()

```
#define AGS_IS_LINK_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_LINK_CHANNEL))
```

AGS_LINK_CHANNEL()

```
#define AGS_LINK_CHANNEL(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_LINK_CHANNEL, AgsLinkChannel))
```

AGS_LINK_CHANNEL_CLASS()

```
#define AGS_LINK_CHANNEL_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_LINK_CHANNEL, AgsLinkChannelClass))
```

AGS_LINK_CHANNEL_GET_CLASS()

```
#define AGS_LINK_CHANNEL_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_LINK_CHANNEL, AgsLinkChannelClass))
```

ags_link_channel_get_type ()

```
GType
ags_link_channel_get_type ();
```

Types and Values

AGS_TYPE_LINK_CHANNEL

```
#define AGS_TYPE_LINK_CHANNEL                   (ags_link_channel_get_type())
```

struct AgsLinkChannel

```
struct AgsLinkChannel;
```

struct AgsLinkChannelClass

```
struct AgsLinkChannelClass {
  AgsTaskClass task;
};
```

Property Details

The "channel" property

```
  "channel"                   AgsChannel~*
```

The assigned AgsChannel

Owner: AgsLinkChannel

Flags: Read / Write

Since: 3.0.0

The "error" property

```
  "error"                     gpointer
```

The assigned GError

Owner: AgsLinkChannel

Flags: Read / Write

Since: 3.0.0

The "link" property

```
  "link"                      AgsChannel~*
```

The assigned AgsChannel link

Owner: AgsLinkChannel

Flags: Read / Write

Since: 3.0.0

24.23 AgsMoveNote

AgsMoveNote — move notation

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_move_note_set_selection ()
AgsMoveNote *	ags_move_note_new ()
#define	AGS_IS_MOVE_NOTE()
#define	AGS_IS_MOVE_NOTE_CLASS()
#define	AGS_MOVE_NOTE()
#define	AGS_MOVE_NOTE_CLASS()
#define	AGS_MOVE_NOTE_GET_CLASS()
GType	ags_move_note_get_type ()

Properties

gboolean	absolute	Read / Write
AgsAudio *	audio	Read / Write
guint	first-x	Read / Write
guint	first-y	Read / Write
int	move-x	Read / Write
int	move-y	Read / Write
AgsNotation *	notation	Read / Write
gboolean	relative	Read / Write
gpointer	selection	Read / Write

Types and Values

#define	AGS_MOVE_NOTE_DEFAULT_X_LENGTH
#define	AGS_MOVE_NOTE_DEFAULT_Y_LENGTH
#define	AGS_TYPE_MOVE_NOTE
struct	AgsMoveNote
struct	AgsMoveNoteClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsMoveNote
```

Includes

```
#include <ags/audio/task/ags_move_note.h>
```

Description

The AgsMoveNote task moves AgsNotation.

Functions

ags_move_note_set_selection ()

```
void
ags_move_note_set_selection (AgsMoveNote *move_note,
                             GList *selection);
```

Set *selection* of *move_note* .

Parameters

move_note	the AgsMoveNote	
selection	the selection as GList.	*[element-type AgsAudio.Note][transfer none]*

Since: 3.2.3

ags_move_note_new ()

```
AgsMoveNote~*
ags_move_note_new (AgsAudio *audio,
                   AgsNotation *notation,
                   GList *selection,
                   guint first_x,
                   guint first_y,
                   gint move_x,
                   gint move_y,
                   gboolean relative,
                   gboolean absolute);
```

Create a new instance of AgsMoveNote task. Note either *relative* or *absolute* shall be TRUE else it won't have any effect.

Parameters

audio	the AgsAudio	
notation	the AgsNotation	
selection	the selection as GList.	*[element-type AgsAudio.Note][transfer none]*
first_x	the x offset to move from	
first_y	the x padding to use	
move_x	the amout to move in x direction	
move_y	the amout to move in y direction	
relative	if TRUE move relative position	
absolute	if TRUE move absolute position	

Returns

a new AgsMoveNote

Since: 3.0.0

AGS_IS_MOVE_NOTE()

```
#define AGS_IS_MOVE_NOTE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_MOVE_NOTE))
```

AGS_IS_MOVE_NOTE_CLASS()

```
#define AGS_IS_MOVE_NOTE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_MOVE_NOTE))
```

AGS_MOVE_NOTE()

```
#define AGS_MOVE_NOTE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MOVE_NOTE, AgsMoveNote))
```

AGS_MOVE_NOTE_CLASS()

```
#define AGS_MOVE_NOTE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_MOVE_NOTE, AgsMoveNoteClass))
```

AGS_MOVE_NOTE_GET_CLASS()

```
#define AGS_MOVE_NOTE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_MOVE_NOTE, AgsMoveNoteClass))
```

ags_move_note_get_type ()

```
GType
ags_move_note_get_type ();
```

Types and Values

AGS_MOVE_NOTE_DEFAULT_X_LENGTH

```
#define AGS_MOVE_NOTE_DEFAULT_X_LENGTH (16 * 16 * 1200)
```

AGS_MOVE_NOTE_DEFAULT_Y_LENGTH

```
#define AGS_MOVE_NOTE_DEFAULT_Y_LENGTH (1024)
```

AGS_TYPE_MOVE_NOTE

```
#define AGS_TYPE_MOVE_NOTE                 (ags_move_note_get_type())
```

struct AgsMoveNote

```
struct AgsMoveNote;
```

struct AgsMoveNoteClass

```
struct AgsMoveNoteClass {
  AgsTaskClass task;
};
```

Property Details

The "absolute" property

```
  "absolute"                  gboolean
```

Move notation by absolute position.

Owner: AgsMoveNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsMoveNote

Flags: Read / Write

Since: 3.0.0

The "first-x" property

```
  "first-x"                   guint
```

Move notation from x offset.

Owner: AgsMoveNote

Flags: Read / Write

Allowed values: <= 307200

Default value: 0

Since: 3.0.0

The "first-y" property

```
  "first-y"                   guint
```

Move notation with x padding.

Owner: AgsMoveNote

Flags: Read / Write

Allowed values: <= 307200

Default value: 0

Since: 3.0.0

The "move-x" property

```
"move-x"                    int
```

Move notation by move-x amount.

Owner: AgsMoveNote

Flags: Read / Write

Allowed values: [-307200,307200]

Default value: 0

Since: 3.0.0

The "move-y" property

```
"move-y"                    int
```

Move notation by move-y amount.

Owner: AgsMoveNote

Flags: Read / Write

Allowed values: [-1024,1024]

Default value: 0

Since: 3.0.0

The "notation" property

```
"notation"                  AgsNotation~*
```

The assigned AgsNotation

Owner: AgsMoveNote

Flags: Read / Write

Since: 3.0.0

The "relative" property

```
"relative"                  gboolean
```

Move notation by relative position.

Owner: AgsMoveNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "selection" property

```
"selection"                gpointer
```

The assigned AgsNote

[type GList(AgsNote)]

Owner: AgsMoveNote

Flags: Read / Write

Since: 3.0.0

24.24 AgsOpenFile

AgsOpenFile — open file task

Stability Level

Stable, unless otherwise indicated

Functions

AgsOpenFile *	ags_open_file_new ()
#define	AGS_IS_OPEN_FILE()
#define	AGS_IS_OPEN_FILE_CLASS()
#define	AGS_OPEN_FILE()
#define	AGS_OPEN_FILE_CLASS()
#define	AGS_OPEN_FILE_GET_CLASS()
GType	ags_open_file_get_type ()

Properties

AgsAudio *	audio	Read / Write
gboolean	create-channels	Read / Write
gpointer	filename	Read / Write
gboolean	overwrite-channels	Read / Write

Types and Values

#define	AGS_TYPE_OPEN_FILE
struct	AgsOpenFile
struct	AgsOpenFileClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsOpenFile
```

Includes

```
#include <ags/audio/task/ags_open_file.h>
```

Description

The AgsOpenFile task opens files.

Functions

ags_open_file_new ()

```
AgsOpenFile~*
ags_open_file_new (AgsAudio *audio,
                   GSList *filename,
                   gboolean overwrite_channels,
                   gboolean create_channels);
```

Create a new instance of AgsOpenFile.

Parameters

audio	the AgsAudio	
filename	the filename to be opened.	*[element-type utf8]*
overwrite_channels	reset existing AgsInput	
create_channels	instantiate new AgsInput	

Returns

the new AgsOpenFile.

Since: 3.0.0

AGS_IS_OPEN_FILE()

```
#define AGS_IS_OPEN_FILE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_OPEN_FILE))
```

AGS_IS_OPEN_FILE_CLASS()

```
#define AGS_IS_OPEN_FILE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_OPEN_FILE))
```

AGS_OPEN_FILE()

```
#define AGS_OPEN_FILE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OPEN_FILE, AgsOpenFile))
```

AGS_OPEN_FILE_CLASS()

```
#define AGS_OPEN_FILE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OPEN_FILE, AgsOpenFileClass))
```

AGS_OPEN_FILE_GET_CLASS()

```
#define AGS_OPEN_FILE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_OPEN_FILE, AgsOpenFileClass))
```

ags_open_file_get_type ()

```
GType
ags_open_file_get_type ();
```

Types and Values

AGS_TYPE_OPEN_FILE

```
#define AGS_TYPE_OPEN_FILE                (ags_open_file_get_type())
```

struct AgsOpenFile

```
struct AgsOpenFile;
```

struct AgsOpenFileClass

```
struct AgsOpenFileClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsOpenFile

Flags: Read / Write

Since: 3.0.0

The "create-channels" property

```
"create-channels"          gboolean
```

As open files create AgsChannel.

Owner: AgsOpenFile

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "filename" property

```
"filename"                 gpointer
```

The assigned GSList providing filename as string

[type GSList(utf8)]

Owner: AgsOpenFile

Flags: Read / Write

Since: 3.0.0

The "overwrite-channels" property

```
"overwrite-channels"       gboolean
```

As open files overwrite AgsChannel.

Owner: AgsOpenFile

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

24.25 AgsOpenSingleFile

AgsOpenSingleFile — open single file

Stability Level

Stable, unless otherwise indicated

Functions

AgsOpenSingleFile *	ags_open_single_file_new ()
#define	AGS_IS_OPEN_SINGLE_FILE()
#define	AGS_IS_OPEN_SINGLE_FILE_CLASS()
#define	AGS_OPEN_SINGLE_FILE()

#define	AGS_OPEN_SINGLE_FILE_CLASS()
#define	AGS_OPEN_SINGLE_FILE_GET_CLASS()
GType	ags_open_single_file_get_type ()

Properties

guint	audio-channel	Read / Write
AgsChannel *	channel	Read / Write
char *	filename	Read / Write

Types and Values

#define	AGS_TYPE_OPEN_SINGLE_FILE
struct	AgsOpenSingleFile
struct	AgsOpenSingleFileClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsOpenSingleFile
```

Includes

```
#include <ags/audio/task/ags_open_single_file.h>
```

Description

The AgsOpenFile task opens one single file.

Functions

ags_open_single_file_new ()

```
AgsOpenSingleFile~*
ags_open_single_file_new (AgsChannel *channel,
                          gchar *filename,
                          guint audio_channel);
```

Create a new instance of AgsOpenSingleFile.

Parameters

channel	the AgsChannel	
filename	the filename to be opened	
audio_channel	the audio channel	

Returns

the new AgsOpenSingleFile.

Since: 3.0.0

AGS_IS_OPEN_SINGLE_FILE()

```
#define AGS_IS_OPEN_SINGLE_FILE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_OPEN_SINGLE_FILE))
```

AGS_IS_OPEN_SINGLE_FILE_CLASS()

```
#define AGS_IS_OPEN_SINGLE_FILE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_OPEN_SINGLE_FILE))
```

AGS_OPEN_SINGLE_FILE()

```
#define AGS_OPEN_SINGLE_FILE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OPEN_SINGLE_FILE, AgsOpenSingleFile))
```

AGS_OPEN_SINGLE_FILE_CLASS()

```
#define AGS_OPEN_SINGLE_FILE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OPEN_SINGLE_FILE, AgsOpenSingleFileClass))
```

AGS_OPEN_SINGLE_FILE_GET_CLASS()

```
#define AGS_OPEN_SINGLE_FILE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_OPEN_SINGLE_FILE, AgsOpenSingleFileClass))
```

ags_open_single_file_get_type ()

```
GType
ags_open_single_file_get_type ();
```

Types and Values

AGS_TYPE_OPEN_SINGLE_FILE

```
#define AGS_TYPE_OPEN_SINGLE_FILE                 (ags_open_single_file_get_type())
```

struct AgsOpenSingleFile

```
struct AgsOpenSingleFile;
```

struct AgsOpenSingleFileClass

```
struct AgsOpenSingleFileClass {
  AgsTaskClass task;
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            guint
```

The audio channel.

Owner: AgsOpenSingleFile

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "channel" property

```
  "channel"                  AgsChannel~*
```

The assigned AgsChannel

Owner: AgsOpenSingleFile

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
  "filename"                 char~*
```

The assigned filename.

Owner: AgsOpenSingleFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

24.26 AgsOpenSf2Instrument

AgsOpenSf2Instrument — open Soundfont2 instrument

Stability Level

Stable, unless otherwise indicated

Functions

AgsOpenSf2Instrument *	ags_open_sf2_instrument_new ()
#define	AGS_IS_OPEN_SF2_INSTRUMENT()
#define	AGS_IS_OPEN_SF2_INSTRUMENT_CLASS()
#define	AGS_OPEN_SF2_INSTRUMENT()
#define	AGS_OPEN_SF2_INSTRUMENT_CLASS()
#define	AGS_OPEN_SF2_INSTRUMENT_GET_CLASS()
GType	ags_open_sf2_instrument_get_type ()

Properties

AgsAudio *	audio	Read / Write
char *	filename	Read / Write
char *	instrument	Read / Write
AgsIpatch *	ipatch	Read / Write
char *	preset	Read / Write
guint	start-pad	Read / Write

Types and Values

#define	AGS_TYPE_OPEN_SF2_INSTRUMENT
struct	AgsOpenSf2Instrument
struct	AgsOpenSf2InstrumentClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsOpenSf2Instrument
```

Includes

```
#include <ags/audio/task/ags_open_sf2_instrument.h>
```

Description

The AgsOpenSf2Instrument task opens Soundfont2 instruments.

Functions

ags_open_sf2_instrument_new ()

```
AgsOpenSf2Instrument~*
ags_open_sf2_instrument_new (AgsAudio *audio,
                             AgsIpatch *ipatch,
                             gchar *filename,
                             gchar *preset,
                             gchar *instrument,
                             guint start_pad);
```

Creates an AgsOpenSf2Instrument.

Parameters

audio	the AgsAudio	
ipatch	the AgsIpatch or NULL	
filename	the Soundfont2 file	
preset	the preset	
instrument	the instrument	
start_pad	the pad start	

Returns

an new AgsOpenSf2Instrument.

Since: 3.0.0

AGS_IS_OPEN_SF2_INSTRUMENT()

```
#define AGS_IS_OPEN_SF2_INSTRUMENT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_OPEN_SF2_INSTRUMENT))
```

AGS_IS_OPEN_SF2_INSTRUMENT_CLASS()

```
#define AGS_IS_OPEN_SF2_INSTRUMENT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_OPEN_SF2_INSTRUMENT))
```

AGS_OPEN_SF2_INSTRUMENT()

```
#define AGS_OPEN_SF2_INSTRUMENT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OPEN_SF2_INSTRUMENT, AgsOpenSf2Instrument))
```

AGS_OPEN_SF2_INSTRUMENT_CLASS()

```
#define AGS_OPEN_SF2_INSTRUMENT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OPEN_SF2_INSTRUMENT, AgsOpenSf2InstrumentClass))
```

AGS_OPEN_SF2_INSTRUMENT_GET_CLASS()

```
#define AGS_OPEN_SF2_INSTRUMENT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_OPEN_SF2_INSTRUMENT, AgsOpenSf2InstrumentClass))
```

ags_open_sf2_instrument_get_type ()

```
GType
ags_open_sf2_instrument_get_type ();
```

Types and Values

AGS_TYPE_OPEN_SF2_INSTRUMENT

```
#define AGS_TYPE_OPEN_SF2_INSTRUMENT                (ags_open_sf2_instrument_get_type())
```

struct AgsOpenSf2Instrument

```
struct AgsOpenSf2Instrument;
```

struct AgsOpenSf2InstrumentClass

```
struct AgsOpenSf2InstrumentClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsOpenSf2Instrument

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
  "filename"                  char~*
```

The assigned filename.

Owner: AgsOpenSf2Instrument

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "instrument" property

```
  "instrument"                char~*
```

The assigned instrument.

Owner: AgsOpenSf2Instrument

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ipatch" property

```
"ipatch"                    AgsIpatch~*
```

The assigned AgsIpatch

Owner: AgsOpenSf2Instrument

Flags: Read / Write

Since: 3.0.0

The "preset" property

```
"preset"                    char~*
```

The assigned preset.

Owner: AgsOpenSf2Instrument

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "start-pad" property

```
"start-pad"                 guint
```

The assigned start-pad.

Owner: AgsOpenSf2Instrument

Flags: Read / Write

Default value: 0

Since: 3.0.0

24.27 AgsOpenSf2Sample

AgsOpenSf2Sample — open Soundfont2 sample

Stability Level

Stable, unless otherwise indicated

Functions

AgsOpenSf2Sample *	ags_open_sf2_sample_new ()
#define	AGS_IS_OPEN_SF2_SAMPLE()
#define	AGS_IS_OPEN_SF2_SAMPLE_CLASS()
#define	AGS_OPEN_SF2_SAMPLE()
#define	AGS_OPEN_SF2_SAMPLE_CLASS()
#define	AGS_OPEN_SF2_SAMPLE_GET_CLASS()

GType	ags_open_sf2_sample_get_type ()

Properties

guint	audio-channel	Read / Write
AgsChannel *	channel	Read / Write
char *	filename	Read / Write
char *	instrument	Read / Write
AgsIpatchSample *	ipatch-sample	Read / Write
char *	preset	Read / Write
char *	sample	Read / Write

Types and Values

#define	AGS_TYPE_OPEN_SF2_SAMPLE
struct	AgsOpenSf2Sample
struct	AgsOpenSf2SampleClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsOpenSf2Sample
```

Includes

```
#include <ags/audio/task/ags_open_sf2_sample.h>
```

Description

The AgsOpenSf2Sample task opens Soundfont2 samples.

Functions

ags_open_sf2_sample_new ()

```
AgsOpenSf2Sample~*
ags_open_sf2_sample_new (AgsChannel *channel,
                         AgsIpatchSample *ipatch_sample,
                         gchar *filename,
                         gchar *preset,
                         gchar *instrument,
                         gchar *sample,
                         guint audio_channel);
```

Creates an AgsOpenSf2Sample.

Parameters

channel	the AgsChannel	
ipatch_sample	the AgsIpatchSample or NULL	
filename	the Soundfont2 file	
preset	the preset	
instrument	the instrument	
sample	the sample	
audio_channel	the audio channel	

Returns

an new AgsOpenSf2Sample.

Since: 3.0.0

AGS_IS_OPEN_SF2_SAMPLE()

```
#define AGS_IS_OPEN_SF2_SAMPLE(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_OPEN_SF2_SAMPLE))
```

AGS_IS_OPEN_SF2_SAMPLE_CLASS()

```
#define AGS_IS_OPEN_SF2_SAMPLE_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_OPEN_SF2_SAMPLE))
```

AGS_OPEN_SF2_SAMPLE()

```
#define AGS_OPEN_SF2_SAMPLE(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OPEN_SF2_SAMPLE, AgsOpenSf2Sample))
```

AGS_OPEN_SF2_SAMPLE_CLASS()

```
#define AGS_OPEN_SF2_SAMPLE_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OPEN_SF2_SAMPLE, AgsOpenSf2SampleClass))
```

AGS_OPEN_SF2_SAMPLE_GET_CLASS()

```
#define AGS_OPEN_SF2_SAMPLE_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_OPEN_SF2_SAMPLE, AgsOpenSf2SampleClass))
```

ags_open_sf2_sample_get_type ()

```
GType
ags_open_sf2_sample_get_type ();
```

Types and Values

AGS_TYPE_OPEN_SF2_SAMPLE

```
#define AGS_TYPE_OPEN_SF2_SAMPLE                   (ags_open_sf2_sample_get_type())
```

struct AgsOpenSf2Sample

```
struct AgsOpenSf2Sample;
```

struct AgsOpenSf2SampleClass

```
struct AgsOpenSf2SampleClass {
  AgsTaskClass task;
};
```

Property Details

The "audio-channel" property

```
  "audio-channel"            guint
```

The nth audio channel.

Owner: AgsOpenSf2Sample

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "channel" property

```
  "channel"                  AgsChannel~*
```

The assigned AgsChannel

Owner: AgsOpenSf2Sample

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
  "filename"                 char~*
```

The assigned filename.

Owner: AgsOpenSf2Sample

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "instrument" property

```
"instrument"                char~*
```

The assigned instrument.

Owner: AgsOpenSf2Sample

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ipatch-sample" property

```
"ipatch-sample"             AgsIpatchSample~*
```

The assigned AgsIpatchSample

Owner: AgsOpenSf2Sample

Flags: Read / Write

Since: 3.0.0

The "preset" property

```
"preset"                    char~*
```

The assigned preset.

Owner: AgsOpenSf2Sample

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "sample" property

```
"sample"                    char~*
```

The assigned sample.

Owner: AgsOpenSf2Sample

Flags: Read / Write

Default value: NULL

Since: 3.0.0

24.28 AgsOpenSFZFile

AgsOpenSFZFile — open SFZ file

Stability Level

Stable, unless otherwise indicated

Functions

AgsOpenSFZFile *	ags_open_sfz_file_new ()
#define	AGS_IS_OPEN_SFZ_FILE()
#define	AGS_IS_OPEN_SFZ_FILE_CLASS()
#define	AGS_OPEN_SFZ_FILE()
#define	AGS_OPEN_SFZ_FILE_CLASS()
#define	AGS_OPEN_SFZ_FILE_GET_CLASS()
GType	ags_open_sfz_file_get_type ()

Properties

AgsAudio *	audio	Read / Write
char *	filename	Read / Write
AgsSFZFile *	sfz-file	Read / Write
guint	start-pad	Read / Write

Types and Values

#define	AGS_TYPE_OPEN_SFZ_FILE
struct	AgsOpenSFZFile
struct	AgsOpenSFZFileClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsOpenSFZFile
```

Includes

```
#include <ags/audio/task/ags_open_sfz_file.h>
```

Description

The AgsOpenSFZFile task opens SFZ file.

Functions

ags_open_sfz_file_new ()

```
AgsOpenSFZFile~*
ags_open_sfz_file_new (AgsAudio *audio,
                       AgsSFZFile *sfz_file,
                       gchar *filename,
                       guint start_pad);
```

Creates an AgsOpenSFZFile.

Parameters

audio	the AgsAudio	
sfz_file	the AgsSFZFile or NULL	
filename	the Soundfont2 file	
start_pad	the pad start	

Returns

an new AgsOpenSFZFile.

Since: 3.4.0

AGS_IS_OPEN_SFZ_FILE()

```
#define AGS_IS_OPEN_SFZ_FILE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_OPEN_SFZ_FILE))
```

AGS_IS_OPEN_SFZ_FILE_CLASS()

```
#define AGS_IS_OPEN_SFZ_FILE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_OPEN_SFZ_FILE))
```

AGS_OPEN_SFZ_FILE()

```
#define AGS_OPEN_SFZ_FILE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OPEN_SFZ_FILE, AgsOpenSFZFile))
```

AGS_OPEN_SFZ_FILE_CLASS()

```
#define AGS_OPEN_SFZ_FILE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OPEN_SFZ_FILE, AgsOpenSFZFileClass))
```

AGS_OPEN_SFZ_FILE_GET_CLASS()

```
#define AGS_OPEN_SFZ_FILE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_OPEN_SFZ_FILE, AgsOpenSFZFileClass))
```

ags_open_sfz_file_get_type ()

```
GType
ags_open_sfz_file_get_type ();
```

Types and Values

AGS_TYPE_OPEN_SFZ_FILE

```
#define AGS_TYPE_OPEN_SFZ_FILE                 (ags_open_sfz_file_get_type())
```

struct AgsOpenSFZFile

```
struct AgsOpenSFZFile;
```

struct AgsOpenSFZFileClass

```
struct AgsOpenSFZFileClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsOpenSFZFile

Flags: Read / Write

Since: 3.4.0

The "filename" property

```
  "filename"                  char~*
```

The assigned filename.

Owner: AgsOpenSFZFile

Flags: Read / Write

Default value: NULL

Since: 3.4.0

The "sfz-file" property

```
  "sfz-file"                  AgsSFZFile~*
```

The assigned AgsSFZFile

Owner: AgsOpenSFZFile

Flags: Read / Write

Since: 3.4.0

The "start-pad" property

```
  "start-pad"                 guint
```

The assigned start-pad.

Owner: AgsOpenSFZFile

Flags: Read / Write

Default value: 0

Since: 3.4.0

24.29 AgsOpenWave

AgsOpenWave — open wave

Stability Level

Stable, unless otherwise indicated

Functions

AgsOpenWave *	ags_open_wave_new ()
#define	AGS_IS_OPEN_WAVE()
#define	AGS_IS_OPEN_WAVE_CLASS()
#define	AGS_OPEN_WAVE()
#define	AGS_OPEN_WAVE_CLASS()
#define	AGS_OPEN_WAVE_GET_CLASS()
GType	ags_open_wave_get_type ()

Properties

AgsAudio *	audio	Read / Write
AgsAudioFile *	audio-file	Read / Write
char *	filename	Read / Write
guint	start-pad	Read / Write

Types and Values

#define	AGS_TYPE_OPEN_WAVE
struct	AgsOpenWave
struct	AgsOpenWaveClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsOpenWave
```

Includes

```
#include <ags/audio/task/ags_open_wave.h>
```

Description

The AgsOpenWave task opens audio files as wave.

Functions

ags_open_wave_new ()

```
AgsOpenWave~*
ags_open_wave_new (AgsAudio *audio,
                   AgsAudioFile *audio_file,
                   gchar *filename,
                   guint start_pad);
```

Creates an AgsOpenWave.

Parameters

audio	the AgsAudio	
audio_file	the AgsAudioFile or NULL	
filename	the Soundfont2 file	
start_pad	the pad start	

Returns

an new AgsOpenWave.

Since: 3.0.0

AGS_IS_OPEN_WAVE()

```
#define AGS_IS_OPEN_WAVE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_OPEN_WAVE))
```

AGS_IS_OPEN_WAVE_CLASS()

```
#define AGS_IS_OPEN_WAVE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_OPEN_WAVE))
```

AGS_OPEN_WAVE()

```
#define AGS_OPEN_WAVE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_OPEN_WAVE, AgsOpenWave))
```

AGS_OPEN_WAVE_CLASS()

```
#define AGS_OPEN_WAVE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_OPEN_WAVE, AgsOpenWaveClass))
```

AGS_OPEN_WAVE_GET_CLASS()

```
#define AGS_OPEN_WAVE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_OPEN_WAVE, AgsOpenWaveClass))
```

ags_open_wave_get_type ()

```
GType
ags_open_wave_get_type ();
```

Types and Values

AGS_TYPE_OPEN_WAVE

```
#define AGS_TYPE_OPEN_WAVE                (ags_open_wave_get_type())
```

struct AgsOpenWave

```
struct AgsOpenWave;
```

struct AgsOpenWaveClass

```
struct AgsOpenWaveClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsOpenWave

Flags: Read / Write

Since: 3.0.0

The "audio-file" property

```
  "audio-file"               AgsAudioFile~*
```

The assigned AgsAudioFile

Owner: AgsOpenWave

Flags: Read / Write

Since: 3.0.0

The "filename" property

```
"filename"                  char~*
```

The assigned filename.

Owner: AgsOpenWave

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "start-pad" property

```
"start-pad"                 guint
```

The assigned start-pad.

Owner: AgsOpenWave

Flags: Read / Write

Default value: 0

Since: 3.0.0

24.30 AgsRemoveAudio

AgsRemoveAudio — remove audio of application context

Stability Level

Stable, unless otherwise indicated

Functions

AgsRemoveAudio *	ags_remove_audio_new ()
#define	AGS_IS_REMOVE_AUDIO()
#define	AGS_IS_REMOVE_AUDIO_CLASS()
#define	AGS_REMOVE_AUDIO()
#define	AGS_REMOVE_AUDIO_CLASS()
#define	AGS_REMOVE_AUDIO_GET_CLASS()
GType	ags_remove_audio_get_type ()

Properties

AgsAudio *	audio	Read / Write

Types and Values

#define	AGS_TYPE_REMOVE_AUDIO
struct	AgsRemoveAudio
struct	AgsRemoveAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsRemoveAudio
```

Includes

```
#include <ags/audio/task/ags_remove_audio.h>
```

Description

The AgsRemoveAudio task removes AgsAudio of AgsApplicationContext.

Functions

ags_remove_audio_new ()

```
AgsRemoveAudio~*
ags_remove_audio_new (AgsAudio *audio);
```

Create a new instance of AgsRemoveAudio.

Parameters

audio	the AgsAudio to remove

Returns

the new AgsRemoveAudio

Since: 3.0.0

AGS_IS_REMOVE_AUDIO()

```
#define AGS_IS_REMOVE_AUDIO(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_REMOVE_AUDIO))
```

AGS_IS_REMOVE_AUDIO_CLASS()

```
#define AGS_IS_REMOVE_AUDIO_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_REMOVE_AUDIO))
```

AGS_REMOVE_AUDIO()

```
#define AGS_REMOVE_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_REMOVE_AUDIO, AgsRemoveAudio))
```

AGS_REMOVE_AUDIO_CLASS()

```
#define AGS_REMOVE_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_REMOVE_AUDIO, AgsRemoveAudioClass))
```

AGS_REMOVE_AUDIO_GET_CLASS()

```
#define AGS_REMOVE_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_REMOVE_AUDIO, AgsRemoveAudioClass))
```

ags_remove_audio_get_type ()

```
GType
ags_remove_audio_get_type ();
```

Types and Values

AGS_TYPE_REMOVE_AUDIO

```
#define AGS_TYPE_REMOVE_AUDIO                (ags_remove_audio_get_type())
```

struct AgsRemoveAudio

```
struct AgsRemoveAudio;
```

struct AgsRemoveAudioClass

```
struct AgsRemoveAudioClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                    AgsAudio~*
```

The assigned AgsAudio

Owner: AgsRemoveAudio

Flags: Read / Write

Since: 3.0.0

24.31 AgsRemoveAudioSignal

AgsRemoveAudioSignal — remove audio_signal object from recycling

Stability Level

Stable, unless otherwise indicated

Functions

AgsRemoveAudioSignal *	ags_remove_audio_signal_new ()
#define	AGS_IS_REMOVE_AUDIO_SIGNAL()
#define	AGS_IS_REMOVE_AUDIO_SIGNAL_CLASS()
#define	AGS_REMOVE_AUDIO_SIGNAL()
#define	AGS_REMOVE_AUDIO_SIGNAL_CLASS()
#define	AGS_REMOVE_AUDIO_SIGNAL_GET_CLASS()
GType	ags_remove_audio_signal_get_type ()

Properties

AgsAudioSignal *	audio-signal	Read / Write
AgsRecycling *	recycling	Read / Write

Types and Values

#define	AGS_TYPE_REMOVE_AUDIO_SIGNAL
struct	AgsRemoveAudioSignal
struct	AgsRemoveAudioSignalClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsRemoveAudioSignal
```

Includes

```
#include <ags/audio/task/ags_remove_audio_signal.h>
```

Description

The AgsRemoveAudioSignal task removes AgsAudioSignal from AgsRecycling.

Functions

ags_remove_audio_signal_new ()

```
AgsRemoveAudioSignal~*
ags_remove_audio_signal_new (AgsRecycling *recycling,
                             AgsAudioSignal *audio_signal);
```

Create a new instance of AgsRemoveAudioSignal.

Parameters

recycling	the AgsRecycling	
audio_signal	the AgsAudioSignal to remove	

Returns

the new AgsRemoveAudioSignal

Since: 3.0.0

AGS_IS_REMOVE_AUDIO_SIGNAL()

```
#define AGS_IS_REMOVE_AUDIO_SIGNAL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_REMOVE_AUDIO_SIGNAL))
```

AGS_IS_REMOVE_AUDIO_SIGNAL_CLASS()

```
#define AGS_IS_REMOVE_AUDIO_SIGNAL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_REMOVE_AUDIO_SIGNAL))
```

AGS_REMOVE_AUDIO_SIGNAL()

```
#define AGS_REMOVE_AUDIO_SIGNAL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_REMOVE_AUDIO_SIGNAL, AgsRemoveAudioSignal))
```

AGS_REMOVE_AUDIO_SIGNAL_CLASS()

```
#define AGS_REMOVE_AUDIO_SIGNAL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_REMOVE_AUDIO_SIGNAL, AgsRemoveAudioSignalClass))
```

AGS_REMOVE_AUDIO_SIGNAL_GET_CLASS()

```
#define AGS_REMOVE_AUDIO_SIGNAL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_REMOVE_AUDIO_SIGNAL, AgsRemoveAudioSignalClass))
```

ags_remove_audio_signal_get_type ()

```
GType
ags_remove_audio_signal_get_type ();
```

Types and Values

AGS_TYPE_REMOVE_AUDIO_SIGNAL

```
#define AGS_TYPE_REMOVE_AUDIO_SIGNAL                (ags_remove_audio_signal_get_type())
```

struct AgsRemoveAudioSignal

```
struct AgsRemoveAudioSignal;
```

struct AgsRemoveAudioSignalClass

```
struct AgsRemoveAudioSignalClass {
  AgsTaskClass task;
};
```

Property Details

The "audio-signal" property

```
  "audio-signal"              AgsAudioSignal~*
```

The assigned AgsAudioSignal

Owner: AgsRemoveAudioSignal

Flags: Read / Write

Since: 3.0.0

The "recycling" property

```
  "recycling"                 AgsRecycling~*
```

The assigned AgsRecycling

Owner: AgsRemoveAudioSignal

Flags: Read / Write

Since: 3.0.0

24.32 AgsRemoveNote

AgsRemoveNote — remove note object from notation

Stability Level

Stable, unless otherwise indicated

Functions

AgsRemoveNote *	ags_remove_note_new ()
#define	AGS_IS_REMOVE_NOTE()
#define	AGS_IS_REMOVE_NOTE_CLASS()
#define	AGS_REMOVE_NOTE()
#define	AGS_REMOVE_NOTE_CLASS()
#define	AGS_REMOVE_NOTE_GET_CLASS()
GType	ags_remove_note_get_type ()

Properties

AgsAudio *	audio	Read / Write
guint	audio-channel	Read / Write
AgsNote *	note	Read / Write
gboolean	use-selection-list	Read / Write

Types and Values

#define	AGS_TYPE_REMOVE_NOTE
struct	AgsRemoveNote
struct	AgsRemoveNoteClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsRemoveNote
```

Includes

```
#include <ags/audio/task/ags_remove_note.h>
```

Description

The AgsRemoveNote task removes AgsNote from AgsNotation.

Functions

ags_remove_note_new ()

```
AgsRemoveNote~*
ags_remove_note_new (AgsAudio *audio,
                     AgsNote *note,
                     guint audio_channel,
                     gboolean use_selection_list);
```

Create a new instance of AgsRemoveNote.

Parameters

audio	the AgsAudio	
note	the AgsNote to remove	
audio_channel	the audio channel	
use_selection_list	if TRUE remove of selection, otherwise of notation	

Returns

the new AgsRemoveNote

Since: 3.0.0

AGS_IS_REMOVE_NOTE()

```
#define AGS_IS_REMOVE_NOTE(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_REMOVE_NOTE))
```

AGS_IS_REMOVE_NOTE_CLASS()

```
#define AGS_IS_REMOVE_NOTE_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_REMOVE_NOTE))
```

AGS_REMOVE_NOTE()

```
#define AGS_REMOVE_NOTE(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_REMOVE_NOTE, AgsRemoveNote))
```

AGS_REMOVE_NOTE_CLASS()

```
#define AGS_REMOVE_NOTE_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_REMOVE_NOTE, AgsRemoveNoteClass))
```

AGS_REMOVE_NOTE_GET_CLASS()

```
#define AGS_REMOVE_NOTE_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_REMOVE_NOTE, AgsRemoveNoteClass))
```

ags_remove_note_get_type ()

```
GType
ags_remove_note_get_type ();
```

Types and Values

AGS_TYPE_REMOVE_NOTE

```
#define AGS_TYPE_REMOVE_NOTE                (ags_remove_note_get_type())
```

struct AgsRemoveNote

```
struct AgsRemoveNote;
```

struct AgsRemoveNoteClass

```
struct AgsRemoveNoteClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsRemoveNote

Flags: Read / Write

Since: 3.0.0

The "audio-channel" property

```
  "audio-channel"             guint
```

The assigned audio channel

Owner: AgsRemoveNote

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "note" property

```
  "note"                      AgsNote~*
```

The assigned AgsNote

Owner: AgsRemoveNote

Flags: Read / Write

Since: 3.0.0

The "use-selection-list" property

```
"use-selection-list"       gboolean
```

The notation's use-selection-list.

Owner: AgsRemoveNote

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

24.33 AgsRemoveSoundcard

AgsRemoveSoundcard — remove soundcard object of application context

Stability Level

Stable, unless otherwise indicated

Functions

AgsRemoveSoundcard *	ags_remove_soundcard_new ()
#define	AGS_IS_REMOVE_SOUNDCARD()
#define	AGS_IS_REMOVE_SOUNDCARD_CLASS()
#define	AGS_REMOVE_SOUNDCARD()
#define	AGS_REMOVE_SOUNDCARD_CLASS()
#define	AGS_REMOVE_SOUNDCARD_GET_CLASS()
GType	ags_remove_soundcard_get_type ()

Properties

GObject *	soundcard	Read / Write

Types and Values

#define	AGS_TYPE_REMOVE_SOUNDCARD
struct	AgsRemoveSoundcard
struct	AgsRemoveSoundcardClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsRemoveSoundcard
```

Includes

```
#include <ags/audio/task/ags_remove_soundcard.h>
```

Description

The AgsRemoveSoundcard task removes AgsSoundcard of application context.

Functions

ags_remove_soundcard_new ()

```
AgsRemoveSoundcard~*
ags_remove_soundcard_new (GObject *soundcard);
```

Create a new instance of AgsRemoveSoundcard.

Parameters

soundcard	the AgsSoundcard to remove

Returns

the new AgsRemoveSoundcard.

Since: 3.0.0

AGS_IS_REMOVE_SOUNDCARD()

```
#define AGS_IS_REMOVE_SOUNDCARD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_REMOVE_SOUNDCARD))
```

AGS_IS_REMOVE_SOUNDCARD_CLASS()

```
#define AGS_IS_REMOVE_SOUNDCARD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_REMOVE_SOUNDCARD))
```

AGS_REMOVE_SOUNDCARD()

```
#define AGS_REMOVE_SOUNDCARD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_REMOVE_SOUNDCARD, AgsRemoveSoundcard))
```

AGS_REMOVE_SOUNDCARD_CLASS()

```
#define AGS_REMOVE_SOUNDCARD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_REMOVE_SOUNDCARD, AgsRemoveSoundcardClass))
```

AGS_REMOVE_SOUNDCARD_GET_CLASS()

```
#define AGS_REMOVE_SOUNDCARD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_REMOVE_SOUNDCARD, AgsRemoveSoundcardClass))
```

ags_remove_soundcard_get_type ()

```
GType
ags_remove_soundcard_get_type ();
```

Types and Values

AGS_TYPE_REMOVE_SOUNDCARD

```
#define AGS_TYPE_REMOVE_SOUNDCARD                (ags_remove_soundcard_get_type())
```

struct AgsRemoveSoundcard

```
struct AgsRemoveSoundcard;
```

struct AgsRemoveSoundcardClass

```
struct AgsRemoveSoundcardClass {
  AgsTaskClass task;
};
```

Property Details

The "soundcard" property

```
  "soundcard"                GObject~*
```

The assigned AgsSoundcard

Owner: AgsRemoveSoundcard

Flags: Read / Write

Since: 3.0.0

24.34 AgsResetNote

AgsResetNote — reset note internal

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_reset_note_add ()
void	ags_reset_note_remove ()
AgsResetNote *	ags_reset_note_get_instance ()
AgsResetNote *	ags_reset_note_new ()
#define	AGS_IS_RESET_NOTE()
#define	AGS_IS_RESET_NOTE_CLASS()
#define	AGS_RESET_NOTE()
#define	AGS_RESET_NOTE_CLASS()
#define	AGS_RESET_NOTE_GET_CLASS()
GType	ags_reset_note_get_type ()

Types and Values

#define	AGS_TYPE_RESET_NOTE
struct	AgsResetNote
struct	AgsResetNoteClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsResetNote
```

Includes

```
#include <ags/audio/task/ags_reset_note.h>
```

Description

The AgsResetNote task resets note to recompute the note during next run.

Functions

ags_reset_note_add ()

```
void
ags_reset_note_add (AgsResetNote *reset_note,
                    AgsNote *note);
```

Add *note* .

Parameters

reset_note	the AgsResetNote	
note	the AgsNote	

Since: 3.0.0

ags_reset_note_remove ()

```
void
ags_reset_note_remove (AgsResetNote *reset_note,
                       AgsNote *note);
```

Remove *note* .

Parameters

reset_note	the AgsResetNote	
note	the AgsNote	

Since: 3.0.0

ags_reset_note_get_instance ()

```
AgsResetNote~*
ags_reset_note_get_instance ();
```

Get instance.

Returns

the AgsResetNote.

[transfer none]

Since: 3.0.0

ags_reset_note_new ()

```
AgsResetNote~*
ags_reset_note_new ();
```

Creates an AgsResetNote.

Returns

an new AgsResetNote.

Since: 3.0.0

AGS_IS_RESET_NOTE()

```
#define AGS_IS_RESET_NOTE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_RESET_NOTE))
```

AGS_IS_RESET_NOTE_CLASS()

```
#define AGS_IS_RESET_NOTE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_RESET_NOTE))
```

AGS_RESET_NOTE()

```
#define AGS_RESET_NOTE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RESET_NOTE, AgsResetNote))
```

AGS_RESET_NOTE_CLASS()

```
#define AGS_RESET_NOTE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RESET_NOTE, AgsResetNoteClass))
```

AGS_RESET_NOTE_GET_CLASS()

```
#define AGS_RESET_NOTE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_RESET_NOTE, AgsResetNoteClass))
```

ags_reset_note_get_type ()

```
GType
ags_reset_note_get_type ();
```

Types and Values

AGS_TYPE_RESET_NOTE

```
#define AGS_TYPE_RESET_NOTE                 (ags_reset_note_get_type())
```

struct AgsResetNote

```
struct AgsResetNote;
```

struct AgsResetNoteClass

```
struct AgsResetNoteClass {
  AgsTaskClass task;
};
```

24.35 AgsResetFxPeak

AgsResetFxPeak — reset fx peak

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_reset_fx_peak_add ()
void	ags_reset_fx_peak_remove ()
AgsResetFxPeak *	ags_reset_fx_peak_get_instance ()
AgsResetFxPeak *	ags_reset_fx_peak_new ()
#define	AGS_IS_RESET_FX_PEAK()
#define	AGS_IS_RESET_FX_PEAK_CLASS()
#define	AGS_RESET_FX_PEAK()
#define	AGS_RESET_FX_PEAK_CLASS()
#define	AGS_RESET_FX_PEAK_GET_CLASS()
GType	ags_reset_fx_peak_get_type ()

Types and Values

#define	AGS_TYPE_RESET_FX_PEAK
struct	AgsResetFxPeak
struct	AgsResetFxPeakClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsResetFxPeak
```

Includes

```
#include <ags/audio/task/ags_reset_fx_peak.h>
```

Description

The AgsResetFxPeak task resets ags-fx-peak to recompute the peak during next run.

Functions

ags_reset_fx_peak_add ()

```
void
ags_reset_fx_peak_add (AgsResetFxPeak *reset_fx_peak,
                       AgsFxPeakChannel *fx_peak_channel);
```

Add *fx_peak_channel* .

Parameters

reset_fx_peak	the AgsResetFxPeak	
fx_peak_channel	the AgsFxPeakChannel	

Since: 3.3.0

ags_reset_fx_peak_remove ()

```
void
ags_reset_fx_peak_remove (AgsResetFxPeak *reset_fx_peak,
                          AgsFxPeakChannel *fx_peak_channel);
```

Remove *fx_peak_channel* .

Parameters

reset_fx_peak	the AgsResetFxPeak	
fx_peak_channel	the AgsFxPeakChannel	

Since: 3.3.0

ags_reset_fx_peak_get_instance ()

```
AgsResetFxPeak~*
ags_reset_fx_peak_get_instance ();
```

Get instance.

Returns

the AgsResetFxPeak.

[transfer none]

Since: 3.3.0

ags_reset_fx_peak_new ()

```
AgsResetFxPeak~*
ags_reset_fx_peak_new ();
```

Creates an AgsResetFxPeak.

Returns

an new AgsResetFxPeak.

Since: 3.3.0

AGS_IS_RESET_FX_PEAK()

```
#define AGS_IS_RESET_FX_PEAK(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_RESET_FX_PEAK))
```

AGS_IS_RESET_FX_PEAK_CLASS()

```
#define AGS_IS_RESET_FX_PEAK_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_RESET_FX_PEAK))
```

AGS_RESET_FX_PEAK()

```
#define AGS_RESET_FX_PEAK(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RESET_FX_PEAK, AgsResetFxPeak))
```

AGS_RESET_FX_PEAK_CLASS()

```
#define AGS_RESET_FX_PEAK_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RESET_FX_PEAK, AgsResetFxPeakClass))
```

AGS_RESET_FX_PEAK_GET_CLASS()

```
#define AGS_RESET_FX_PEAK_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_RESET_FX_PEAK, AgsResetFxPeakClass))
```

ags_reset_fx_peak_get_type ()

```
GType
ags_reset_fx_peak_get_type ();
```

Types and Values

AGS_TYPE_RESET_FX_PEAK

```
#define AGS_TYPE_RESET_FX_PEAK                (ags_reset_fx_peak_get_type())
```

struct AgsResetFxPeak

```
struct AgsResetFxPeak;
```

struct AgsResetFxPeakClass

```
struct AgsResetFxPeakClass {
  AgsTaskClass task;
};
```

24.36 AgsResetFxAnalyse

AgsResetFxAnalyse — reset fx analyse

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_reset_fx_analyse_add ()
void	ags_reset_fx_analyse_remove ()
AgsResetFxAnalyse *	ags_reset_fx_analyse_get_instance ()
AgsResetFxAnalyse *	ags_reset_fx_analyse_new ()
#define	AGS_IS_RESET_FX_ANALYSE()
#define	AGS_IS_RESET_FX_ANALYSE_CLASS()
#define	AGS_RESET_FX_ANALYSE()
#define	AGS_RESET_FX_ANALYSE_CLASS()
#define	AGS_RESET_FX_ANALYSE_GET_CLASS()
GType	ags_reset_fx_analyse_get_type ()

Types and Values

#define	AGS_TYPE_RESET_FX_ANALYSE
struct	AgsResetFxAnalyse
struct	AgsResetFxAnalyseClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsResetFxAnalyse
```

Includes

```
#include <ags/audio/task/ags_reset_fx_analyse.h>
```

Description

The AgsResetFxAnalyse task resets ags-fx-analyse to recompute the analyse during next run.

Functions

ags_reset_fx_analyse_add ()

```
void
ags_reset_fx_analyse_add (AgsResetFxAnalyse *reset_fx_analyse,
                          AgsFxAnalyseChannel *fx_analyse_channel);
```

Add *fx_analyse_channel* .

Parameters

reset_fx_analyse	the AgsResetFxAnalyse	
fx_analyse_channel	the AgsFxAnalyseChannel	

Since: 3.3.0

ags_reset_fx_analyse_remove ()

```
void
ags_reset_fx_analyse_remove (AgsResetFxAnalyse *reset_fx_analyse,
                             AgsFxAnalyseChannel *fx_analyse_channel);
```

Remove *fx_analyse_channel* .

Parameters

reset_fx_analyse	the AgsResetFxAnalyse	
fx_analyse_channel	the AgsFxAnalyseChannel	

Since: 3.3.0

ags_reset_fx_analyse_get_instance ()

```
AgsResetFxAnalyse~*
ags_reset_fx_analyse_get_instance ();
```

Get instance.

Returns

the AgsResetFxAnalyse.

[transfer none]

Since: 3.3.0

ags_reset_fx_analyse_new ()

```
AgsResetFxAnalyse~*
ags_reset_fx_analyse_new ();
```

Creates an AgsResetFxAnalyse.

Returns

an new AgsResetFxAnalyse.

Since: 3.3.0

AGS_IS_RESET_FX_ANALYSE()

```
#define AGS_IS_RESET_FX_ANALYSE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_RESET_FX_ANALYSE))
```

AGS_IS_RESET_FX_ANALYSE_CLASS()

```
#define AGS_IS_RESET_FX_ANALYSE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_RESET_FX_ANALYSE))
```

AGS_RESET_FX_ANALYSE()

```
#define AGS_RESET_FX_ANALYSE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RESET_FX_ANALYSE, AgsResetFxAnalyse))
```

AGS_RESET_FX_ANALYSE_CLASS()

```
#define AGS_RESET_FX_ANALYSE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RESET_FX_ANALYSE, AgsResetFxAnalyseClass))
```

AGS_RESET_FX_ANALYSE_GET_CLASS()

```
#define AGS_RESET_FX_ANALYSE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_RESET_FX_ANALYSE, AgsResetFxAnalyseClass))
```

ags_reset_fx_analyse_get_type ()

```
GType
ags_reset_fx_analyse_get_type ();
```

Types and Values

AGS_TYPE_RESET_FX_ANALYSE

```
#define AGS_TYPE_RESET_FX_ANALYSE                (ags_reset_fx_analyse_get_type())
```

struct AgsResetFxAnalyse

```
struct AgsResetFxAnalyse;
```

struct AgsResetFxAnalyseClass

```
struct AgsResetFxAnalyseClass {
  AgsTaskClass task;
};
```

24.37 AgsResizeAudio

AgsResizeAudio — resize audio task

Stability Level

Stable, unless otherwise indicated

Functions

AgsResizeAudio *	ags_resize_audio_new ()
#define	AGS_IS_RESIZE_AUDIO()
#define	AGS_IS_RESIZE_AUDIO_CLASS()
#define	AGS_RESIZE_AUDIO()
#define	AGS_RESIZE_AUDIO_CLASS()
#define	AGS_RESIZE_AUDIO_GET_CLASS()
GType	ags_resize_audio_get_type ()

Properties

AgsAudio *	audio	Read / Write
guint	audio-channels	Read / Write
guint	input-pads	Read / Write
guint	output-pads	Read / Write

Types and Values

#define	AGS_TYPE_RESIZE_AUDIO
struct	AgsResizeAudio
struct	AgsResizeAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsResizeAudio
```

Includes

```
#include <ags/audio/task/ags_resize_audio.h>
```

Description

The AgsResizeAudio task resizes AgsAudio.

Functions

ags_resize_audio_new ()

```
AgsResizeAudio~*
ags_resize_audio_new (AgsAudio *audio,
                      guint output_pads,
                      guint input_pads,
                      guint audio_channels);
```

Creates an AgsResizeAudio.

Parameters

audio	the AgsAudio to resize	
output_pads	output pads	
input_pads	input pads	
audio_channels	audio channels	

Returns

an new AgsResizeAudio.

Since: 3.0.0

AGS_IS_RESIZE_AUDIO()

```
#define AGS_IS_RESIZE_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_RESIZE_AUDIO))
```

AGS_IS_RESIZE_AUDIO_CLASS()

```
#define AGS_IS_RESIZE_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_RESIZE_AUDIO))
```

AGS_RESIZE_AUDIO()

```
#define AGS_RESIZE_AUDIO(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RESIZE_AUDIO, AgsResizeAudio))
```

AGS_RESIZE_AUDIO_CLASS()

```
#define AGS_RESIZE_AUDIO_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_RESIZE_AUDIO, AgsResizeAudioClass))
```

AGS_RESIZE_AUDIO_GET_CLASS()

```
#define AGS_RESIZE_AUDIO_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_RESIZE_AUDIO, AgsResizeAudioClass))
```

ags_resize_audio_get_type ()

```
GType
ags_resize_audio_get_type ();
```

Types and Values

AGS_TYPE_RESIZE_AUDIO

```
#define AGS_TYPE_RESIZE_AUDIO                   (ags_resize_audio_get_type())
```

struct AgsResizeAudio

```
struct AgsResizeAudio;
```

struct AgsResizeAudioClass

```
struct AgsResizeAudioClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsResizeAudio

Flags: Read / Write

Since: 3.0.0

The "audio-channels" property

```
  "audio-channels"            guint
```

The count of audio channels to apply to audio.

Owner: AgsResizeAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "input-pads" property

```
  "input-pads"                guint
```

The count of input pads to apply to audio.

Owner: AgsResizeAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "output-pads" property

```
"output-pads"              guint
```

The count of output pads to apply to audio.

Owner: AgsResizeAudio

Flags: Read / Write

Default value: 0

Since: 3.0.0

24.38 AgsSeekSoundcard

AgsSeekSoundcard — seek soundcard object

Stability Level

Stable, unless otherwise indicated

Functions

AgsSeekSoundcard *	ags_seek_soundcard_new ()
#define	AGS_IS_SEEK_SOUNDCARD()
#define	AGS_IS_SEEK_SOUNDCARD_CLASS()
#define	AGS_SEEK_SOUNDCARD()
#define	AGS_SEEK_SOUNDCARD_CLASS()
#define	AGS_SEEK_SOUNDCARD_GET_CLASS()
GType	ags_seek_soundcard_get_type ()

Properties

gint64	offset	Read / Write
GObject *	soundcard	Read / Write
guint	whence	Read / Write

Types and Values

#define	AGS_TYPE_SEEK_SOUNDCARD
struct	AgsSeekSoundcard
struct	AgsSeekSoundcardClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsSeekSoundcard
```

Includes

```
#include <ags/audio/task/ags_seek_soundcard.h>
```

Description

The AgsSeekSoundcard task seeks AgsSoundcard.

Functions

ags_seek_soundcard_new ()

```
AgsSeekSoundcard~*
ags_seek_soundcard_new (GObject *soundcard,
                        gint64 offset,
                        guint whence);
```

Create a new instance of AgsSeekSoundcard.

Parameters

soundcard	the GObject sub-type implementing AgsSoundcard	
offset	the offset	
whence	whence see AgsSeekType	

Returns

the new AgsSeekSoundcard

Since: 3.0.0

AGS_IS_SEEK_SOUNDCARD()

```
#define AGS_IS_SEEK_SOUNDCARD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SEEK_SOUNDCARD))
```

AGS_IS_SEEK_SOUNDCARD_CLASS()

```
#define AGS_IS_SEEK_SOUNDCARD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SEEK_SOUNDCARD))
```

AGS_SEEK_SOUNDCARD()

```
#define AGS_SEEK_SOUNDCARD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SEEK_SOUNDCARD, AgsSeekSoundcard))
```

AGS_SEEK_SOUNDCARD_CLASS()

```
#define AGS_SEEK_SOUNDCARD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SEEK_SOUNDCARD, AgsSeekSoundcardClass))
```

AGS_SEEK_SOUNDCARD_GET_CLASS()

```
#define AGS_SEEK_SOUNDCARD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SEEK_SOUNDCARD, AgsSeekSoundcardClass))
```

ags_seek_soundcard_get_type ()

```
GType
ags_seek_soundcard_get_type ();
```

Types and Values

AGS_TYPE_SEEK_SOUNDCARD

```
#define AGS_TYPE_SEEK_SOUNDCARD                 (ags_seek_soundcard_get_type())
```

struct AgsSeekSoundcard

```
struct AgsSeekSoundcard;
```

struct AgsSeekSoundcardClass

```
struct AgsSeekSoundcardClass {
  AgsTaskClass task;
};
```

Property Details

The "offset" property

```
  "offset"                    gint64
```

The offset to seek.

Owner: AgsSeekSoundcard

Flags: Read / Write

Allowed values: >= 0

Default value: 0

Since: 3.0.0

The "soundcard" property

```
"soundcard"                 GObject~*
```

The assigned AgsSoundcard

Owner: AgsSeekSoundcard

Flags: Read / Write

Since: 3.0.0

The "whence" property

```
"whence"                    guint
```

Whence either AGS_SEEK_SET, AGS_SEEK_CUR or AGS_SEEK_END

Owner: AgsSeekSoundcard

Flags: Read / Write

Default value: 1

Since: 3.0.0

24.39 AgsSetAudioChannels

AgsSetAudioChannels — resizes audio channels

Stability Level

Stable, unless otherwise indicated

Functions

AgsSetAudioChannels *	ags_set_audio_channels_new ()
#define	AGS_IS_SET_AUDIO_CHANNELS()
#define	AGS_IS_SET_AUDIO_CHANNELS_CLASS()
#define	AGS_SET_AUDIO_CHANNELS()
#define	AGS_SET_AUDIO_CHANNELS_CLASS()
#define	AGS_SET_AUDIO_CHANNELS_GET_CLASS()
GType	ags_set_audio_channels_get_type ()

Properties

guint	audio-channels	Read / Write
GObject *	soundcard	Read / Write

Types and Values

#define	AGS_TYPE_SET_AUDIO_CHANNELS
struct	AgsSetAudioChannels
struct	AgsSetAudioChannelsClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsSetAudioChannels
```

Includes

```
#include <ags/audio/task/ags_set_audio_channels.h>
```

Description

The AgsSetAudioChannels task resizes audio channels of AgsSoundcard.

Functions

ags_set_audio_channels_new ()

```
AgsSetAudioChannels~*
ags_set_audio_channels_new (GObject *soundcard,
                            guint audio_channels);
```

Create a new instance of AgsSetAudioChannels.

Parameters

soundcard	the AgsSoundcard to reset	
audio_channels	the new count of audio channels	

Returns

the new AgsSetAudioChannels.

Since: 3.0.0

AGS_IS_SET_AUDIO_CHANNELS()

```
#define AGS_IS_SET_AUDIO_CHANNELS(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SET_AUDIO_CHANNELS))
```

AGS_IS_SET_AUDIO_CHANNELS_CLASS()

```
#define AGS_IS_SET_AUDIO_CHANNELS_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SET_AUDIO_CHANNELS))
```

AGS_SET_AUDIO_CHANNELS()

```
#define AGS_SET_AUDIO_CHANNELS(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SET_AUDIO_CHANNELS, AgsSetAudioChannels))
```

AGS_SET_AUDIO_CHANNELS_CLASS()

```
#define AGS_SET_AUDIO_CHANNELS_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SET_AUDIO_CHANNELS, AgsSetAudioChannelsClass))
```

AGS_SET_AUDIO_CHANNELS_GET_CLASS()

```
#define AGS_SET_AUDIO_CHANNELS_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SET_AUDIO_CHANNELS, AgsSetAudioChannelsClass))
```

ags_set_audio_channels_get_type ()

```
GType
ags_set_audio_channels_get_type ();
```

Types and Values

AGS_TYPE_SET_AUDIO_CHANNELS

```
#define AGS_TYPE_SET_AUDIO_CHANNELS                 (ags_set_audio_channels_get_type())
```

struct AgsSetAudioChannels

```
struct AgsSetAudioChannels;
```

struct AgsSetAudioChannelsClass

```
struct AgsSetAudioChannelsClass {
  AgsTaskClass task;
};
```

Property Details

The "audio-channels" property

```
"audio-channels"            guint
```

The count of audio channels to apply to soundcard.

Owner: AgsSetAudioChannels

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "soundcard" property

```
"soundcard"                 GObject~*
```

The assigned AgsSoundcard instance.

Owner: AgsSetAudioChannels

Flags: Read / Write

Since: 3.0.0

24.40 AgsSetBufferSize

AgsSetBufferSize — adjust buffer size

Stability Level

Stable, unless otherwise indicated

Functions

AgsSetBufferSize *	ags_set_buffer_size_new ()
#define	AGS_IS_SET_BUFFER_SIZE()
#define	AGS_IS_SET_BUFFER_SIZE_CLASS()
#define	AGS_SET_BUFFER_SIZE()
#define	AGS_SET_BUFFER_SIZE_CLASS()
#define	AGS_SET_BUFFER_SIZE_GET_CLASS()
GType	ags_set_buffer_size_get_type ()

Properties

guint	buffer-size	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_SET_BUFFER_SIZE
struct	AgsSetBufferSize
struct	AgsSetBufferSizeClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsSetBufferSize
```

Includes

```
#include <ags/audio/task/ags_set_buffer_size.h>
```

Description

The AgsSetBufferSize task adjusts buffer size of scope.

Functions

ags_set_buffer_size_new ()

```
AgsSetBufferSize~*
ags_set_buffer_size_new (GObject *scope,
                         guint buffer_size);
```

Create a new instance of AgsSetBufferSize.

Parameters

scope	the GObject to reset	
buffer_size	the new buffer size	

Returns

the new AgsSetBufferSize

Since: 3.0.0

AGS_IS_SET_BUFFER_SIZE()

```
#define AGS_IS_SET_BUFFER_SIZE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SET_BUFFER_SIZE))
```

AGS_IS_SET_BUFFER_SIZE_CLASS()

```
#define AGS_IS_SET_BUFFER_SIZE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SET_BUFFER_SIZE))
```

AGS_SET_BUFFER_SIZE()

```
#define AGS_SET_BUFFER_SIZE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SET_BUFFER_SIZE, AgsSetBufferSize))
```

AGS_SET_BUFFER_SIZE_CLASS()

```
#define AGS_SET_BUFFER_SIZE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SET_BUFFER_SIZE, AgsSetBufferSizeClass))
```

AGS_SET_BUFFER_SIZE_GET_CLASS()

```
#define AGS_SET_BUFFER_SIZE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SET_BUFFER_SIZE, AgsSetBufferSizeClass))
```

ags_set_buffer_size_get_type ()

```
GType
ags_set_buffer_size_get_type ();
```

Types and Values

AGS_TYPE_SET_BUFFER_SIZE

```
#define AGS_TYPE_SET_BUFFER_SIZE                (ags_set_buffer_size_get_type())
```

struct AgsSetBufferSize

```
struct AgsSetBufferSize;
```

struct AgsSetBufferSizeClass

```
struct AgsSetBufferSizeClass {
  AgsTaskClass task;
};
```

Property Details

The "buffer-size" property

```
  "buffer-size"              guint
```

The buffer size to apply to scope.

Owner: AgsSetBufferSize

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "scope" property

```
"scope"                    GObject~*
```

The assigned GObject as scope.

Owner: AgsSetBufferSize

Flags: Read / Write

Since: 3.0.0

24.41 AgsSetDevice

AgsSetDevice — set device

Stability Level

Stable, unless otherwise indicated

Functions

AgsSetDevice *	ags_set_device_new ()
#define	AGS_IS_SET_DEVICE()
#define	AGS_IS_SET_DEVICE_CLASS()
#define	AGS_SET_DEVICE()
#define	AGS_SET_DEVICE_CLASS()
#define	AGS_SET_DEVICE_GET_CLASS()
GType	ags_set_device_get_type ()

Properties

char *	device	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_SET_DEVICE
struct	AgsSetDevice
struct	AgsSetDeviceClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTask
    &#x2570;&#x2500;&#x2500; AgsSetDevice
```

Includes

```
#include <ags/audio/task/ags_set_device.h>
```

Description

The AgsSetDevice task sets device of AgsSoundcard or AgsSequencer.

Functions

ags_set_device_new ()

```
AgsSetDevice~*
ags_set_device_new (GObject *scope,
                    gchar *device);
```

Create a new instance of AgsSetDevice.

Parameters

scope	the AgsSoundcard or AgsSequencer to reset	
device	the device as string	

Returns

the new AgsSetDevice.

Since: 3.0.0

AGS_IS_SET_DEVICE()

```
#define AGS_IS_SET_DEVICE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SET_DEVICE))
```

AGS_IS_SET_DEVICE_CLASS()

```
#define AGS_IS_SET_DEVICE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SET_DEVICE))
```

AGS_SET_DEVICE()

```
#define AGS_SET_DEVICE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SET_DEVICE, AgsSetDevice))
```

AGS_SET_DEVICE_CLASS()

```
#define AGS_SET_DEVICE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SET_DEVICE, AgsSetDeviceClass))
```

AGS_SET_DEVICE_GET_CLASS()

```
#define AGS_SET_DEVICE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SET_DEVICE, AgsSetDeviceClass))
```

ags_set_device_get_type ()

```
GType
ags_set_device_get_type ();
```

Types and Values

AGS_TYPE_SET_DEVICE

```
#define AGS_TYPE_SET_DEVICE                 (ags_set_device_get_type())
```

struct AgsSetDevice

```
struct AgsSetDevice;
```

struct AgsSetDeviceClass

```
struct AgsSetDeviceClass {
  AgsTaskClass task;
};
```

Property Details

The "device" property

```
  "device"                   char~*
```

The device to set.

Owner: AgsSetDevice

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "scope" property

```
  "scope"                    GObject~*
```

The assigned AgsSoundcard or AgsSequencer instance.

Owner: AgsSetDevice

Flags: Read / Write

Since: 3.0.0

24.42 AgsSetFormat

AgsSetFormat — reset format

Stability Level

Stable, unless otherwise indicated

Functions

AgsSetFormat *	ags_set_format_new ()
#define	AGS_IS_SET_FORMAT()
#define	AGS_IS_SET_FORMAT_CLASS()
#define	AGS_SET_FORMAT()
#define	AGS_SET_FORMAT_CLASS()
#define	AGS_SET_FORMAT_GET_CLASS()
GType	ags_set_format_get_type ()

Properties

guint	format	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_SET_FORMAT
struct	AgsSetFormat
struct	AgsSetFormatClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsSetFormat
```

Includes

```
#include <ags/audio/task/ags_set_format.h>
```

Description

The AgsSetFormat task resets format of scope.

Functions

ags_set_format_new ()

```
AgsSetFormat~*
ags_set_format_new (GObject *scope,
                    guint format);
```

Create a new instance of AgsSetFormat.

Parameters

scope	the GObject to reset	
format	the new format	

Returns

the new AgsSetFormat

Since: 3.0.0

AGS_IS_SET_FORMAT()

```
#define AGS_IS_SET_FORMAT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SET_FORMAT))
```

AGS_IS_SET_FORMAT_CLASS()

```
#define AGS_IS_SET_FORMAT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SET_FORMAT))
```

AGS_SET_FORMAT()

```
#define AGS_SET_FORMAT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SET_FORMAT, AgsSetFormat))
```

AGS_SET_FORMAT_CLASS()

```
#define AGS_SET_FORMAT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SET_FORMAT, AgsSetFormatClass))
```

AGS_SET_FORMAT_GET_CLASS()

```
#define AGS_SET_FORMAT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SET_FORMAT, AgsSetFormatClass))
```

ags_set_format_get_type ()

```
GType
ags_set_format_get_type ();
```

Types and Values

AGS_TYPE_SET_FORMAT

```
#define AGS_TYPE_SET_FORMAT                 (ags_set_format_get_type())
```

struct AgsSetFormat

```
struct AgsSetFormat;
```

struct AgsSetFormatClass

```
struct AgsSetFormatClass {
  AgsTaskClass task;
};
```

Property Details

The "format" property

```
  "format"                    guint
```

The format to apply to scope.

Owner: AgsSetFormat

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "scope" property

```
  "scope"                     GObject~*
```

The assigned GObject as scope.

Owner: AgsSetFormat

Flags: Read / Write

Since: 3.0.0

24.43 AgsSetMuted

AgsSetMuted — set muted

Stability Level

Stable, unless otherwise indicated

Functions

AgsSetMuted *	ags_set_muted_new ()
#define	AGS_IS_SET_MUTED()
#define	AGS_IS_SET_MUTED_CLASS()
#define	AGS_SET_MUTED()
#define	AGS_SET_MUTED_CLASS()
#define	AGS_SET_MUTED_GET_CLASS()
GType	ags_set_muted_get_type ()

Properties

gboolean	muted	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_SET_MUTED
struct	AgsSetMuted
struct	AgsSetMutedClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsSetMuted
```

Includes

```
#include <ags/audio/task/recall/ags_set_muted.h>
```

Description

The AgsSetMuted task sets muted to AgsMutable.

Functions

ags_set_muted_new ()

```
AgsSetMuted~*
ags_set_muted_new (GObject *scope,
                   gboolean muted);
```

Creates an AgsSetMuted.

Parameters

scope	the GObject	
muted	muted to set	

Returns

an new AgsSetMuted.

Since: 3.0.0

AGS_IS_SET_MUTED()

```
#define AGS_IS_SET_MUTED(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SET_MUTED))
```

AGS_IS_SET_MUTED_CLASS()

```
#define AGS_IS_SET_MUTED_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SET_MUTED))
```

AGS_SET_MUTED()

```
#define AGS_SET_MUTED(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SET_MUTED, AgsSetMuted))
```

AGS_SET_MUTED_CLASS()

```
#define AGS_SET_MUTED_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SET_MUTED, AgsSetMutedClass))
```

AGS_SET_MUTED_GET_CLASS()

```
#define AGS_SET_MUTED_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SET_MUTED, AgsSetMutedClass))
```

ags_set_muted_get_type ()

```
GType
ags_set_muted_get_type ();
```

Types and Values

AGS_TYPE_SET_MUTED

```
#define AGS_TYPE_SET_MUTED                 (ags_set_muted_get_type())
```

struct AgsSetMuted

```
struct AgsSetMuted;
```

struct AgsSetMutedClass

```
struct AgsSetMutedClass {
  AgsTaskClass task;
};
```

Property Details

The "muted" property

```
  "muted"                    gboolean
```

The muted to set to scope.

Owner: AgsSetMuted

Flags: Read / Write

Default value: FALSE

Since: 3.0.0

The "scope" property

```
  "scope"                    GObject~*
```

The assigned GObject as scope.

Owner: AgsSetMuted

Flags: Read / Write

Since: 3.0.0

24.44 AgsSetSamplerate

AgsSetSamplerate — modify samplerate

Stability Level

Stable, unless otherwise indicated

Functions

AgsSetSamplerate *	ags_set_samplerate_new ()
#define	AGS_IS_SET_SAMPLERATE()
#define	AGS_IS_SET_SAMPLERATE_CLASS()
#define	AGS_SET_SAMPLERATE()
#define	AGS_SET_SAMPLERATE_CLASS()
#define	AGS_SET_SAMPLERATE_GET_CLASS()
GType	ags_set_samplerate_get_type ()

Properties

guint	samplerate	Read / Write
GObject *	scope	Read / Write

Types and Values

#define	AGS_TYPE_SET_SAMPLERATE
struct	AgsSetSamplerate
struct	AgsSetSamplerateClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsSetSamplerate
```

Includes

```
#include <ags/audio/task/ags_set_samplerate.h>
```

Description

The AgsSetSamplerate task modifies samplerate of scope.

Functions

ags_set_samplerate_new ()

```
AgsSetSamplerate~*
ags_set_samplerate_new (GObject *scope,
                        guint samplerate);
```

Create a new instance of AgsSetSamplerate.

Parameters

scope	the AgsSoundcard to reset	
samplerate	the new samplerate	

Returns

the new AgsSetSamplerate

Since: 3.0.0

AGS_IS_SET_SAMPLERATE()

```
#define AGS_IS_SET_SAMPLERATE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SET_SAMPLERATE))
```

AGS_IS_SET_SAMPLERATE_CLASS()

```
#define AGS_IS_SET_SAMPLERATE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SET_SAMPLERATE))
```

AGS_SET_SAMPLERATE()

```
#define AGS_SET_SAMPLERATE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SET_SAMPLERATE, AgsSetSamplerate))
```

AGS_SET_SAMPLERATE_CLASS()

```
#define AGS_SET_SAMPLERATE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SET_SAMPLERATE, AgsSetSamplerateClass))
```

AGS_SET_SAMPLERATE_GET_CLASS()

```
#define AGS_SET_SAMPLERATE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SET_SAMPLERATE, AgsSetSamplerateClass))
```

ags_set_samplerate_get_type ()

```
GType
ags_set_samplerate_get_type ();
```

Types and Values

AGS_TYPE_SET_SAMPLERATE

```
#define AGS_TYPE_SET_SAMPLERATE                 (ags_set_samplerate_get_type())
```

struct AgsSetSamplerate

```
struct AgsSetSamplerate;
```

struct AgsSetSamplerateClass

```
struct AgsSetSamplerateClass {
  AgsTaskClass task;
};
```

Property Details

The "samplerate" property

```
  "samplerate"                guint
```

The samplerate to apply to scope.

Owner: AgsSetSamplerate

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "scope" property

```
  "scope"                     GObject~*
```

The assigned GObject as scope.

Owner: AgsSetSamplerate

Flags: Read / Write

Since: 3.0.0

24.45 AgsStartAudio

AgsStartAudio — start audio object to audio loop

Stability Level

Stable, unless otherwise indicated

Functions

AgsStartAudio *	ags_start_audio_new ()
#define	AGS_IS_START_AUDIO()
#define	AGS_IS_START_AUDIO_CLASS()
#define	AGS_START_AUDIO()
#define	AGS_START_AUDIO_CLASS()
#define	AGS_START_AUDIO_GET_CLASS()
GType	ags_start_audio_get_type ()

Properties

AgsAudio *	audio	Read / Write
int	sound-scope	Read / Write

Types and Values

#define	AGS_TYPE_START_AUDIO
struct	AgsStartAudio
struct	AgsStartAudioClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStartAudio
```

Includes

```
#include <ags/audio/task/ags_start_audio.h>
```

Description

The AgsStartAudio task starts AgsAudio to AgsAudioLoop.

Functions

ags_start_audio_new ()

```
AgsStartAudio~*
ags_start_audio_new (AgsAudio *audio,
                     gint sound_scope);
```

Create a new instance of AgsStartAudio.

Parameters

audio	the AgsAudio to start	
sound_scope	the AgsSoundScope or -1 for all	

Returns

the new AgsStartAudio.

Since: 3.0.0

AGS_IS_START_AUDIO()

```
#define AGS_IS_START_AUDIO(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_START_AUDIO))
```

AGS_IS_START_AUDIO_CLASS()

```
#define AGS_IS_START_AUDIO_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_START_AUDIO))
```

AGS_START_AUDIO()

```
#define AGS_START_AUDIO(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_START_AUDIO, AgsStartAudio))
```

AGS_START_AUDIO_CLASS()

```
#define AGS_START_AUDIO_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_START_AUDIO, AgsStartAudioClass))
```

AGS_START_AUDIO_GET_CLASS()

```
#define AGS_START_AUDIO_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_START_AUDIO, AgsStartAudioClass))
```

ags_start_audio_get_type ()

```
GType
ags_start_audio_get_type ();
```

Types and Values

AGS_TYPE_START_AUDIO

```
#define AGS_TYPE_START_AUDIO                (ags_start_audio_get_type())
```

struct AgsStartAudio

```
struct AgsStartAudio;
```

struct AgsStartAudioClass

```
struct AgsStartAudioClass {
  AgsTaskClass task;
};
```

Property Details

The "audio" property

```
  "audio"                     AgsAudio~*
```

The assigned AgsAudio

Owner: AgsStartAudio

Flags: Read / Write

Since: 3.0.0

The "sound-scope" property

```
  "sound-scope"               int
```

The effects sound-scope.

Owner: AgsStartAudio

Flags: Read / Write

Allowed values: [-1,5]

Default value: -1

Since: 3.0.0

24.46 AgsStartChannel

AgsStartChannel — start channel object to audio loop

Stability Level

Stable, unless otherwise indicated

Functions

AgsStartChannel *	ags_start_channel_new ()
#define	AGS_IS_START_CHANNEL()
#define	AGS_IS_START_CHANNEL_CLASS()
#define	AGS_START_CHANNEL()
#define	AGS_START_CHANNEL_CLASS()
#define	AGS_START_CHANNEL_GET_CLASS()
GType	ags_start_channel_get_type ()

Properties

AgsChannel *	channel	Read / Write
int	sound-scope	Read / Write

Types and Values

#define	AGS_TYPE_START_CHANNEL
struct	AgsStartChannel
struct	AgsStartChannelClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStartChannel
```

Includes

```
#include <ags/audio/task/ags_start_channel.h>
```

Description

The AgsStartChannel task starts AgsChannel to AgsAudioLoop.

Functions

ags_start_channel_new ()

```
AgsStartChannel~*
ags_start_channel_new (AgsChannel *channel,
                       gint sound_scope);
```

Create a new instance of AgsStartChannel.

Parameters

channel	the AgsChannel to start	
sound_scope	the sound scope	

Returns

the new AgsStartChannel.

Since: 3.0.0

AGS_IS_START_CHANNEL()

```
#define AGS_IS_START_CHANNEL(obj)                (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_START_CHANNEL))
```

AGS_IS_START_CHANNEL_CLASS()

```
#define AGS_IS_START_CHANNEL_CLASS(class)        (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_START_CHANNEL))
```

AGS_START_CHANNEL()

```
#define AGS_START_CHANNEL(obj)                   (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_START_CHANNEL, AgsStartChannel))
```

AGS_START_CHANNEL_CLASS()

```
#define AGS_START_CHANNEL_CLASS(class)           (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_START_CHANNEL, AgsStartChannelClass))
```

AGS_START_CHANNEL_GET_CLASS()

```
#define AGS_START_CHANNEL_GET_CLASS(obj)         (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_START_CHANNEL, AgsStartChannelClass))
```

ags_start_channel_get_type ()

```
GType
ags_start_channel_get_type ();
```

Types and Values

AGS_TYPE_START_CHANNEL

```
#define AGS_TYPE_START_CHANNEL                   (ags_start_channel_get_type())
```

struct AgsStartChannel

```
struct AgsStartChannel;
```

struct AgsStartChannelClass

```
struct AgsStartChannelClass {
  AgsTaskClass task;
};
```

Property Details

The "channel" property

```
  "channel"                   AgsChannel~*
```

The assigned AgsChannel

Owner: AgsStartChannel

Flags: Read / Write

Since: 3.0.0

The "sound-scope" property

```
  "sound-scope"               int
```

The effects sound-scope.

Owner: AgsStartChannel

Flags: Read / Write

Allowed values: [-1,5]

Default value: -1

Since: 3.0.0

24.47 AgsStartSequencer

AgsStartSequencer — start sequencer object

Stability Level

Stable, unless otherwise indicated

Functions

AgsStartSequencer *	ags_start_sequencer_new ()
#define	AGS_IS_START_SEQUENCER()
#define	AGS_IS_START_SEQUENCER_CLASS()
#define	AGS_START_SEQUENCER()
#define	AGS_START_SEQUENCER_CLASS()
#define	AGS_START_SEQUENCER_GET_CLASS()
GType	ags_start_sequencer_get_type ()

Types and Values

#define	AGS_TYPE_START_SEQUENCER
struct	AgsStartSequencer
struct	AgsStartSequencerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStartSequencer
```

Includes

```
#include <ags/audio/task/ags_start_sequencer.h>
```

Description

The AgsStartSequencer task starts sequencer.

Functions

ags_start_sequencer_new ()

```
AgsStartSequencer~*
ags_start_sequencer_new ();
```

Creates an AgsStartSequencer.

Returns

an new AgsStartSequencer.

Since: 3.0.0

AGS_IS_START_SEQUENCER()

```
#define AGS_IS_START_SEQUENCER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_START_SEQUENCER))
```

AGS_IS_START_SEQUENCER_CLASS()

```
#define AGS_IS_START_SEQUENCER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_START_SEQUENCER))
```

AGS_START_SEQUENCER()

```
#define AGS_START_SEQUENCER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_START_SEQUENCER, AgsStartSequencer))
```

AGS_START_SEQUENCER_CLASS()

```
#define AGS_START_SEQUENCER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_START_SEQUENCER, AgsStartSequencerClass))
```

AGS_START_SEQUENCER_GET_CLASS()

```
#define AGS_START_SEQUENCER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_START_SEQUENCER, AgsStartSequencerClass))
```

ags_start_sequencer_get_type ()

```
GType
ags_start_sequencer_get_type ();
```

Types and Values

AGS_TYPE_START_SEQUENCER

```
#define AGS_TYPE_START_SEQUENCER                 (ags_start_sequencer_get_type())
```

struct AgsStartSequencer

```
struct AgsStartSequencer;
```

struct AgsStartSequencerClass

```
struct AgsStartSequencerClass {
  AgsTaskClass task;
};
```

24.48 AgsStartSoundcard

AgsStartSoundcard — start soundcard object

Stability Level

Stable, unless otherwise indicated

Functions

AgsStartSoundcard *	ags_start_soundcard_new ()
#define	AGS_IS_START_SOUNDCARD()
#define	AGS_IS_START_SOUNDCARD_CLASS()
#define	AGS_START_SOUNDCARD()
#define	AGS_START_SOUNDCARD_CLASS()
#define	AGS_START_SOUNDCARD_GET_CLASS()
GType	ags_start_soundcard_get_type ()

Types and Values

#define	AGS_TYPE_START_SOUNDCARD
struct	AgsStartSoundcard
struct	AgsStartSoundcardClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStartSoundcard
```

Includes

```
#include <ags/audio/task/ags_start_soundcard.h>
```

Description

The AgsStartSoundcard task starts soundcard.

Functions

ags_start_soundcard_new ()

```
AgsStartSoundcard~*
ags_start_soundcard_new ();
```

Create a new instance of AgsStartSoundcard.

Returns

the new AgsStartSoundcard.

Since: 3.0.0

AGS_IS_START_SOUNDCARD()

```
#define AGS_IS_START_SOUNDCARD(obj)             (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_START_SOUNDCARD))
```

AGS_IS_START_SOUNDCARD_CLASS()

```
#define AGS_IS_START_SOUNDCARD_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_START_SOUNDCARD))
```

AGS_START_SOUNDCARD()

```
#define AGS_START_SOUNDCARD(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_START_SOUNDCARD, AgsStartSoundcard))
```

AGS_START_SOUNDCARD_CLASS()

```
#define AGS_START_SOUNDCARD_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_START_SOUNDCARD, AgsStartSoundcardClass))
```

AGS_START_SOUNDCARD_GET_CLASS()

```
#define AGS_START_SOUNDCARD_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_START_SOUNDCARD, AgsStartSoundcardClass))
```

ags_start_soundcard_get_type ()

```
GType
ags_start_soundcard_get_type ();
```

Types and Values

AGS_TYPE_START_SOUNDCARD

```
#define AGS_TYPE_START_SOUNDCARD                (ags_start_soundcard_get_type())
```

struct AgsStartSoundcard

```
struct AgsStartSoundcard;
```

struct AgsStartSoundcardClass

```
struct AgsStartSoundcardClass {
  AgsTaskClass task;
};
```

24.49 AgsStopSequencer

AgsStopSequencer — stop sequencer object

Stability Level

Stable, unless otherwise indicated

Functions

AgsStopSequencer *	ags_stop_sequencer_new ()
#define	AGS_IS_STOP_SEQUENCER()
#define	AGS_IS_STOP_SEQUENCER_CLASS()
#define	AGS_STOP_SEQUENCER()
#define	AGS_STOP_SEQUENCER_CLASS()
#define	AGS_STOP_SEQUENCER_GET_CLASS()
GType	ags_stop_sequencer_get_type ()

Types and Values

#define	AGS_TYPE_STOP_SEQUENCER
struct	AgsStopSequencer
struct	AgsStopSequencerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStopSequencer
```

Includes

```
#include <ags/audio/task/ags_stop_sequencer.h>
```

Description

The AgsStopSequencer task stops sequencer.

Functions

ags_stop_sequencer_new ()

```
AgsStopSequencer~*
ags_stop_sequencer_new ();
```

Creates an AgsStopSequencer.

Returns

an new AgsStopSequencer.

Since: 3.0.0

AGS_IS_STOP_SEQUENCER()

```
#define AGS_IS_STOP_SEQUENCER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_STOP_SEQUENCER))
```

AGS_IS_STOP_SEQUENCER_CLASS()

```
#define AGS_IS_STOP_SEQUENCER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_STOP_SEQUENCER))
```

AGS_STOP_SEQUENCER()

```
#define AGS_STOP_SEQUENCER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_STOP_SEQUENCER, AgsStopSequencer))
```

AGS_STOP_SEQUENCER_CLASS()

```
#define AGS_STOP_SEQUENCER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_STOP_SEQUENCER, AgsStopSequencerClass))
```

AGS_STOP_SEQUENCER_GET_CLASS()

```
#define AGS_STOP_SEQUENCER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_STOP_SEQUENCER, AgsStopSequencerClass))
```

ags_stop_sequencer_get_type ()

```
GType
ags_stop_sequencer_get_type ();
```

Types and Values

AGS_TYPE_STOP_SEQUENCER

```
#define AGS_TYPE_STOP_SEQUENCER                 (ags_stop_sequencer_get_type())
```

struct AgsStopSequencer

```
struct AgsStopSequencer;
```

struct AgsStopSequencerClass

```
struct AgsStopSequencerClass {
  AgsTaskClass task;
};
```

24.50 AgsStopSoundcard

AgsStopSoundcard — stop soundcard object

Stability Level

Stable, unless otherwise indicated

Functions

AgsStopSoundcard *	ags_stop_soundcard_new ()
#define	AGS_IS_STOP_SOUNDCARD()
#define	AGS_IS_STOP_SOUNDCARD_CLASS()
#define	AGS_STOP_SOUNDCARD()
#define	AGS_STOP_SOUNDCARD_CLASS()
#define	AGS_STOP_SOUNDCARD_GET_CLASS()
GType	ags_stop_soundcard_get_type ()

Types and Values

#define	AGS_TYPE_STOP_SOUNDCARD
struct	AgsStopSoundcard
struct	AgsStopSoundcardClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStopSoundcard
```

Includes

```
#include <ags/audio/task/ags_stop_soundcard.h>
```

Description

The AgsStopSoundcard task stops soundcard.

Functions

ags_stop_soundcard_new ()

```
AgsStopSoundcard~*
ags_stop_soundcard_new ();
```

Create a new instance of AgsStopSoundcard.

Returns

the new AgsStopSoundcard.

Since: 3.0.0

AGS_IS_STOP_SOUNDCARD()

```
#define AGS_IS_STOP_SOUNDCARD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_STOP_SOUNDCARD))
```

AGS_IS_STOP_SOUNDCARD_CLASS()

```
#define AGS_IS_STOP_SOUNDCARD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_STOP_SOUNDCARD))
```

AGS_STOP_SOUNDCARD()

```
#define AGS_STOP_SOUNDCARD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_STOP_SOUNDCARD, AgsStopSoundcard))
```

AGS_STOP_SOUNDCARD_CLASS()

```
#define AGS_STOP_SOUNDCARD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_STOP_SOUNDCARD, AgsStopSoundcardClass))
```

AGS_STOP_SOUNDCARD_GET_CLASS()

```
#define AGS_STOP_SOUNDCARD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_STOP_SOUNDCARD, AgsStopSoundcardClass))
```

ags_stop_soundcard_get_type ()

```
GType
ags_stop_soundcard_get_type ();
```

Types and Values

AGS_TYPE_STOP_SOUNDCARD

```
#define AGS_TYPE_STOP_SOUNDCARD                (ags_stop_soundcard_get_type())
```

struct AgsStopSoundcard

```
struct AgsStopSoundcard;
```

struct AgsStopSoundcardClass

```
struct AgsStopSoundcardClass {
  AgsTaskClass task;
};
```

24.51 AgsStopThread

AgsStopThread — stop all threads

Stability Level

Stable, unless otherwise indicated

Functions

AgsStopThread *	ags_stop_thread_new ()
#define	AGS_IS_STOP_THREAD()
#define	AGS_IS_STOP_THREAD_CLASS()
#define	AGS_STOP_THREAD()
#define	AGS_STOP_THREAD_CLASS()
#define	AGS_STOP_THREAD_GET_CLASS()
GType	ags_stop_thread_get_type ()

Types and Values

#define	AGS_TYPE_STOP_THREAD
struct	AgsStopThread
struct	AgsStopThreadClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsStopThread
```

Includes

```
#include <ags/audio/task/ags_stop_thread.h>
```

Description

The AgsStopThread task stops all threads.

Functions

ags_stop_thread_new ()

```
AgsStopThread~*
ags_stop_thread_new ();
```

Creates a new instance of AgsStopThread.

Returns

the new AgsStopThread.

Since: 3.16.10

AGS_IS_STOP_THREAD()

```
#define AGS_IS_STOP_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_STOP_THREAD))
```

AGS_IS_STOP_THREAD_CLASS()

```
#define AGS_IS_STOP_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_STOP_THREAD))
```

AGS_STOP_THREAD()

```
#define AGS_STOP_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_STOP_THREAD, AgsStopThread))
```

AGS_STOP_THREAD_CLASS()

```
#define AGS_STOP_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_STOP_THREAD, AgsStopThreadClass))
```

AGS_STOP_THREAD_GET_CLASS()

```
#define AGS_STOP_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_STOP_THREAD, AgsStopThreadClass))
```

ags_stop_thread_get_type ()

```
GType
ags_stop_thread_get_type ();
```

Types and Values

AGS_TYPE_STOP_THREAD

```
#define AGS_TYPE_STOP_THREAD                (ags_stop_thread_get_type())
```

struct AgsStopThread

```
struct AgsStopThread;
```

struct AgsStopThreadClass

```
struct AgsStopThreadClass {
  AgsTaskClass task;
};
```

24.52 AgsSwitchBufferFlag

AgsSwitchBufferFlag — switch buffer flag of device

Stability Level

Stable, unless otherwise indicated

Functions

AgsSwitchBufferFlag *	ags_switch_buffer_flag_new ()
#define	AGS_IS_SWITCH_BUFFER_FLAG()
#define	AGS_IS_SWITCH_BUFFER_FLAG_CLASS()
#define	AGS_SWITCH_BUFFER_FLAG()
#define	AGS_SWITCH_BUFFER_FLAG_CLASS()
#define	AGS_SWITCH_BUFFER_FLAG_GET_CLASS()
GType	ags_switch_buffer_flag_get_type ()

Properties

GObject *	device	Read / Write

Types and Values

#define	AGS_TYPE_SWITCH_BUFFER_FLAG
struct	AgsSwitchBufferFlag
struct	AgsSwitchBufferFlagClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsSwitchBufferFlag
```

Includes

```
#include <ags/audio/task/ags_switch_buffer_flag.h>
```

Description

The AgsSwitchBufferFlag task switches the buffer flag of device.

Functions

ags_switch_buffer_flag_new ()

```
AgsSwitchBufferFlag~*
ags_switch_buffer_flag_new (GObject *device);
```

Creates an AgsSwitchBufferFlag.

Parameters

device	the AgsSoundcard or AgsSequencer

Returns

an new AgsSwitchBufferFlag.

Since: 3.0.0

AGS_IS_SWITCH_BUFFER_FLAG()

```
#define AGS_IS_SWITCH_BUFFER_FLAG(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SWITCH_BUFFER_FLAG))
```

AGS_IS_SWITCH_BUFFER_FLAG_CLASS()

```
#define AGS_IS_SWITCH_BUFFER_FLAG_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_SWITCH_BUFFER_FLAG))
```

AGS_SWITCH_BUFFER_FLAG()

```
#define AGS_SWITCH_BUFFER_FLAG(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SWITCH_BUFFER_FLAG, AgsSwitchBufferFlag))
```

AGS_SWITCH_BUFFER_FLAG_CLASS()

```
#define AGS_SWITCH_BUFFER_FLAG_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SWITCH_BUFFER_FLAG, AgsSwitchBufferFlagClass))
```

AGS_SWITCH_BUFFER_FLAG_GET_CLASS()

```
#define AGS_SWITCH_BUFFER_FLAG_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_SWITCH_BUFFER_FLAG, AgsSwitchBufferFlagClass))
```

ags_switch_buffer_flag_get_type ()

```
GType
ags_switch_buffer_flag_get_type ();
```

Types and Values

AGS_TYPE_SWITCH_BUFFER_FLAG

```
#define AGS_TYPE_SWITCH_BUFFER_FLAG                 (ags_switch_buffer_flag_get_type())
```

struct AgsSwitchBufferFlag

```
struct AgsSwitchBufferFlag;
```

struct AgsSwitchBufferFlagClass

```
struct AgsSwitchBufferFlagClass {
  AgsTaskClass task;
};
```

Property Details

The "device" property

```
  "device"                    GObject~*
```

The assigned AgsSoundcard or AgsSequencer

Owner: AgsSwitchBufferFlag

Flags: Read / Write

Since: 3.0.0

24.53 AgsTicDevice

AgsTicDevice — tic device object

Stability Level

Stable, unless otherwise indicated

Functions

AgsTicDevice *	ags_tic_device_new ()
#define	AGS_IS_TIC_DEVICE()
#define	AGS_IS_TIC_DEVICE_CLASS()
#define	AGS_TIC_DEVICE()
#define	AGS_TIC_DEVICE_CLASS()
#define	AGS_TIC_DEVICE_GET_CLASS()
GType	ags_tic_device_get_type ()

Properties

GObject *	device	Read / Write

Types and Values

#define	AGS_TYPE_TIC_DEVICE
struct	AgsTicDevice
struct	AgsTicDeviceClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsTicDevice
```

Includes

```
#include <ags/audio/task/ags_tic_device.h>
```

Description

The AgsTicDevice task tics GObject.

Functions

ags_tic_device_new ()

```
AgsTicDevice~*
ags_tic_device_new (GObject *device);
```

Creates an AgsTicDevice.

Parameters

device	the GObject to tic

Returns

an new AgsTicDevice.

Since: 3.0.0

AGS_IS_TIC_DEVICE()

```
#define AGS_IS_TIC_DEVICE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_TIC_DEVICE))
```

AGS_IS_TIC_DEVICE_CLASS()

```
#define AGS_IS_TIC_DEVICE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_TIC_DEVICE))
```

AGS_TIC_DEVICE()

```
#define AGS_TIC_DEVICE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TIC_DEVICE, AgsTicDevice))
```

AGS_TIC_DEVICE_CLASS()

```
#define AGS_TIC_DEVICE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_TIC_DEVICE, AgsTicDeviceClass))
```

AGS_TIC_DEVICE_GET_CLASS()

```
#define AGS_TIC_DEVICE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_TIC_DEVICE, AgsTicDeviceClass))
```

ags_tic_device_get_type ()

```
GType
ags_tic_device_get_type ();
```

Types and Values

AGS_TYPE_TIC_DEVICE

```
#define AGS_TYPE_TIC_DEVICE                 (ags_tic_device_get_type())
```

struct AgsTicDevice

```
struct AgsTicDevice;
```

struct AgsTicDeviceClass

```
struct AgsTicDeviceClass {
  AgsTaskClass task;
};
```

Property Details

The "device" property

```
  "device"                    GObject~*
```

The assigned GObject as device.

Owner: AgsTicDevice

Flags: Read / Write

Since: 3.0.0

24.54 AgsTogglePatternBit

AgsTogglePatternBit — toggle the pattern

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_toggle_pattern_bit_refresh_gui ()
AgsTogglePatternBit *	ags_toggle_pattern_bit_new ()
#define	AGS_IS_TOGGLE_PATTERN_BIT()
#define	AGS_IS_TOGGLE_PATTERN_BIT_CLASS()
#define	AGS_TOGGLE_PATTERN_BIT()
#define	AGS_TOGGLE_PATTERN_BIT_CLASS()
#define	AGS_TOGGLE_PATTERN_BIT_GET_CLASS()
GType	ags_toggle_pattern_bit_get_type ()

Properties

guint	bit	Read / Write
guint	index-i	Read / Write
guint	index-j	Read / Write
guint	line	Read / Write

AgsPattern *	pattern	Read / Write

Types and Values

#define	AGS_TYPE_TOGGLE_PATTERN_BIT
struct	AgsTogglePatternBit
struct	AgsTogglePatternBitClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
        &#x2570;&#x2500;&#x2500; AgsTogglePatternBit
```

Includes

```
#include <ags/audio/task/ags_toggle_pattern_bit.h>
```

Description

The AgsTogglePatternBit task toggles the specified AgsPattern.

Functions

ags_toggle_pattern_bit_refresh_gui ()

```
void
ags_toggle_pattern_bit_refresh_gui (AgsTogglePatternBit *toggle_pattern_bit);
```

ags_toggle_pattern_bit_new ()

```
AgsTogglePatternBit~*
ags_toggle_pattern_bit_new (AgsPattern *pattern,
                            guint line,
                            guint index_i,
                            guint index_j,
                            guint bit);
```

Creates an AgsTogglePatternBit.

Parameters

pattern	the AgsPattern to toggle	
line	the affected line	
index_i	bank 0	
index_j	bank 1	
bit	the index within pattern	

Returns

an new AgsTogglePatternBit.

Since: 3.0.0

AGS_IS_TOGGLE_PATTERN_BIT()

```
#define AGS_IS_TOGGLE_PATTERN_BIT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_TOGGLE_PATTERN_BIT))
```

AGS_IS_TOGGLE_PATTERN_BIT_CLASS()

```
#define AGS_IS_TOGGLE_PATTERN_BIT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_TOGGLE_PATTERN_BIT))
```

AGS_TOGGLE_PATTERN_BIT()

```
#define AGS_TOGGLE_PATTERN_BIT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TOGGLE_PATTERN_BIT, AgsTogglePatternBit))
```

AGS_TOGGLE_PATTERN_BIT_CLASS()

```
#define AGS_TOGGLE_PATTERN_BIT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_TOGGLE_PATTERN_BIT, AgsTogglePatternBitClass))
```

AGS_TOGGLE_PATTERN_BIT_GET_CLASS()

```
#define AGS_TOGGLE_PATTERN_BIT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_TOGGLE_PATTERN_BIT, AgsTogglePatternBitClass))
```

ags_toggle_pattern_bit_get_type ()

```
GType
ags_toggle_pattern_bit_get_type ();
```

Types and Values

AGS_TYPE_TOGGLE_PATTERN_BIT

```
#define AGS_TYPE_TOGGLE_PATTERN_BIT                 (ags_toggle_pattern_bit_get_type())
```

struct AgsTogglePatternBit

```
struct AgsTogglePatternBit;
```

struct AgsTogglePatternBitClass

```
struct AgsTogglePatternBitClass {
  AgsTaskClass task;

  void (*refresh_gui)(AgsTogglePatternBit *toggle_pattern_bit);
};
```

Property Details

The "bit" property

```
  "bit"                       guint
```

The bit.

Owner: AgsTogglePatternBit

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "index-i" property

```
  "index-i"                   guint
```

The index-i.

Owner: AgsTogglePatternBit

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "index-j" property

```
  "index-j"                   guint
```

The index-j.

Owner: AgsTogglePatternBit

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "line" property

```
  "line"                      guint
```

The line.

Owner: AgsTogglePatternBit

Flags: Read / Write

Default value: 0

Since: 3.0.0

The "pattern" property

```
"pattern"                    AgsPattern~*
```

The assigned AgsPattern

Owner: AgsTogglePatternBit

Flags: Read / Write

Since: 3.0.0

Part III

Annotation Glossary

A

allow-none

NULL is OK, both for passing and for returning.

array

Parameter points to an array of items.

E

element-type

Generics and defining elements of containers and arrays.

I

inout

Parameter for input and for returning results. Default is transfer full.

N

nullable

NULL may be passed as the value in, out, in-out; or as a return value.

O

out

Parameter for returning results. Default is transfer full.

out caller-allocates

Out parameter, where caller must allocate storage.

S

scope call

The callback is valid only during the call to the method.

Stable

The intention of a Stable interface is to enable arbitrary third parties to develop applications to these interfaces, release them, and have confidence that they will run on all minor releases of the product (after the one in which the interface was introduced, and within the same major release). Even at a major release, incompatible changes are expected to be rare, and to have strong justifications.

T

transfer container

The caller owns the data container, but not the data inside it.

transfer full

The caller owns the data, and is responsible for free it.

transfer none

The data is owned by the callee, which is responsible of freeing it.

type

Override the parsed C type with given type.

Part IV

Tree Index

```
GObject
&#x251c;&#x2500;&#x2500; AgsAcceleration
&#x251c;&#x2500;&#x2500; AgsTask
&#x2502;   &#x251c;&#x2500;&#x2500; AgsAddAudio
&#x2502;   &#x251c;&#x2500;&#x2500; AgsAddAudioSignal
&#x2502;   &#x251c;&#x2500;&#x2500; AgsAddNote
&#x2502;   &#x251c;&#x2500;&#x2500; AgsAddSoundcard
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplyBpm
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplyPresets
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySequencerLength
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySF2MidiLocale
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySF2Synth
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySFZInstrument
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySFZSynth
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySoundConfig
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplySynth
&#x2502;   &#x251c;&#x2500;&#x2500; AgsApplyTact
&#x2502;   &#x251c;&#x2500;&#x2500; AgsCancelAudio
&#x2502;   &#x251c;&#x2500;&#x2500; AgsCancelChannel
&#x2502;   &#x251c;&#x2500;&#x2500; AgsClearAudioSignal
&#x2502;   &#x251c;&#x2500;&#x2500; AgsClearBuffer
&#x2502;   &#x251c;&#x2500;&#x2500; AgsCropNote
&#x2502;   &#x251c;&#x2500;&#x2500; AgsExportOutput
&#x2502;   &#x251c;&#x2500;&#x2500; AgsFreeSelection
&#x2502;   &#x251c;&#x2500;&#x2500; AgsLinkChannel
&#x2502;   &#x251c;&#x2500;&#x2500; AgsMoveNote
&#x2502;   &#x251c;&#x2500;&#x2500; AgsOpenFile
&#x2502;   &#x251c;&#x2500;&#x2500; AgsOpenSf2Instrument
&#x2502;   &#x251c;&#x2500;&#x2500; AgsOpenSf2Sample
&#x2502;   &#x251c;&#x2500;&#x2500; AgsOpenSFZFile
&#x2502;   &#x251c;&#x2500;&#x2500; AgsOpenSingleFile
&#x2502;   &#x251c;&#x2500;&#x2500; AgsOpenWave
&#x2502;   &#x251c;&#x2500;&#x2500; AgsRemoveAudio
&#x2502;   &#x251c;&#x2500;&#x2500; AgsRemoveAudioSignal
&#x2502;   &#x251c;&#x2500;&#x2500; AgsRemoveNote
&#x2502;   &#x251c;&#x2500;&#x2500; AgsRemoveSoundcard
&#x2502;   &#x251c;&#x2500;&#x2500; AgsResetFxAnalyse
&#x2502;   &#x251c;&#x2500;&#x2500; AgsResetFxPeak
&#x2502;   &#x251c;&#x2500;&#x2500; AgsResetNote
&#x2502;   &#x251c;&#x2500;&#x2500; AgsResizeAudio
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSeekSoundcard
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSetAudioChannels
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSetBufferSize
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSetDevice
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSetFormat
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSetMuted
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSetSamplerate
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStartAudio
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStartChannel
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStartSequencer
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStartSoundcard
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStopSequencer
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStopSoundcard
&#x2502;   &#x251c;&#x2500;&#x2500; AgsStopThread
&#x2502;   &#x251c;&#x2500;&#x2500; AgsSwitchBufferFlag
&#x2502;   &#x251c;&#x2500;&#x2500; AgsTicDevice
&#x2502;   &#x2570;&#x2500;&#x2500; AgsTogglePatternBit
&#x251c;&#x2500;&#x2500; AgsAlsaDevin
&#x251c;&#x2500;&#x2500; AgsAlsaDevout
&#x251c;&#x2500;&#x2500; AgsAlsaMidiin
&#x251c;&#x2500;&#x2500; AgsApplicationContext
```

```
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsAudioApplicationContext
    &#x251c;&#x2500;&#x2500; AgsAudioContainer
    &#x251c;&#x2500;&#x2500; AgsAudioContainerManager
    &#x251c;&#x2500;&#x2500; AgsAudioFile
    &#x251c;&#x2500;&#x2500; AgsFileLink
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsAudioFileLink
    &#x251c;&#x2500;&#x2500; AgsAudioFileManager
    &#x251c;&#x2500;&#x2500; AgsAudio
    &#x251c;&#x2500;&#x2500; AgsThread
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsAudioLoop
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsAudioThread
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsChannelThread
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsExportThread
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsSequencerThread
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsSoundcardThread
    &#x251c;&#x2500;&#x2500; AgsAudioSignal
    &#x251c;&#x2500;&#x2500; AgsAudioUnitClient
    &#x251c;&#x2500;&#x2500; AgsAudioUnitDevin
    &#x251c;&#x2500;&#x2500; AgsAudioUnitDevout
    &#x251c;&#x2500;&#x2500; AgsAudioUnitPort
    &#x251c;&#x2500;&#x2500; AgsAudioUnitServer
    &#x251c;&#x2500;&#x2500; AgsAutomation
    &#x251c;&#x2500;&#x2500; AgsBasePlugin
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsDssiPlugin
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsLadspaPlugin
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsLv2Plugin
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsLv2uiPlugin
    &#x251c;&#x2500;&#x2500; AgsBuffer
    &#x251c;&#x2500;&#x2500; AgsChannel
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsInput
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsOutput
    &#x251c;&#x2500;&#x2500; AgsCoreAudioClient
    &#x251c;&#x2500;&#x2500; AgsCoreAudioDevin
    &#x251c;&#x2500;&#x2500; AgsCoreAudioDevout
    &#x251c;&#x2500;&#x2500; AgsCoreAudioMidiin
    &#x251c;&#x2500;&#x2500; AgsCoreAudioPort
    &#x251c;&#x2500;&#x2500; AgsCoreAudioServer
    &#x251c;&#x2500;&#x2500; AgsDssiManager
    &#x251c;&#x2500;&#x2500; AgsFifoout
    &#x251c;&#x2500;&#x2500; AgsFrequencyMap
    &#x251c;&#x2500;&#x2500; AgsFrequencyMapManager
    &#x251c;&#x2500;&#x2500; AgsRecall
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsRecallAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxAnalyseAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxBufferAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxNotationAudio
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxDssiAudio
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxFMSynthAudio
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLv2Audio
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSF2SynthAudio
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSFZSynthAudio
    &#x2502;   &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxSynthAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEnvelopeAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEq10Audio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxHighPassAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLadspaAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLfoAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLowPassAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPatternAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPeakAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPlaybackAudio
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseAudio
```

```
    &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxVolumeAudio
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsRecallAudioRun
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxAnalyseAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxBufferAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxNotationAudioProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxDssiAudioProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxFMSynthAudioProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLv2AudioProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSF2SynthAudioProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSFZSynthAudioProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxSynthAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEnvelopeAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEq10AudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxHighPassAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLadspaAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLfoAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLowPassAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPatternAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPeakAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPlaybackAudioProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseAudioProcessor
    &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxVolumeAudioProcessor
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsRecallAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxAnalyseAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxBufferAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxNotationAudioSignal
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxDssiAudioSignal
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxFMSynthAudioSignal
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLv2AudioSignal
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSF2SynthAudioSignal
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSFZSynthAudioSignal
    &#x2502;   &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxSynthAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEnvelopeAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEq10AudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxHighPassAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLadspaAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLfoAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLowPassAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPatternAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPeakAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPlaybackAudioSignal
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseAudioSignal
    &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxVolumeAudioSignal
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsRecallChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxAnalyseChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxBufferChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxNotationChannel
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxDssiChannel
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxFMSynthChannel
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLv2Channel
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSF2SynthChannel
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSFZSynthChannel
    &#x2502;   &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxSynthChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEnvelopeChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEq10Channel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxHighPassChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLadspaChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLfoChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLowPassChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPatternChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPeakChannel
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPlaybackChannel
```

```
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseChannel
    &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxVolumeChannel
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsRecallChannelRun
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxAnalyseChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxBufferChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxNotationChannelProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxDssiChannelProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxFMSynthChannelProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLv2ChannelProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSF2SynthChannelProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSFZSynthChannelProcessor
    &#x2502;   &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxSynthChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEnvelopeChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxEq10ChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxHighPassChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLadspaChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLfoChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLowPassChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPatternChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPeakChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxPlaybackChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseChannelProcessor
    &#x2502;   &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxVolumeChannelProcessor
    &#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsGenericRecallChannelRun
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsRecallRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxAnalyseRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxBufferRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxNotationRecycling
    &#x2502;       &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxDssiRecycling
    &#x2502;       &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxFMSynthRecycling
    &#x2502;       &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxLv2Recycling
    &#x2502;       &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSF2SynthRecycling
    &#x2502;       &#x2502;   &#x251c;&#x2500;&#x2500; AgsFxSFZSynthRecycling
    &#x2502;       &#x2502;   &#x2570;&#x2500;&#x2500; AgsFxSynthRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxEnvelopeRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxEq10Recycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxHighPassRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxLadspaRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxLfoRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxLowPassRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxPatternRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxPeakRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxPlaybackRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxTwoPassAliaseRecycling
    &#x2502;       &#x251c;&#x2500;&#x2500; AgsFxVolumeRecycling
    &#x2502;       &#x2570;&#x2500;&#x2500; AgsGenericRecallRecycling
    &#x251c;&#x2500;&#x2500; AgsFxFactory
    &#x251c;&#x2500;&#x2500; AgsGstreamerClient
    &#x251c;&#x2500;&#x2500; AgsGstreamerDevin
    &#x251c;&#x2500;&#x2500; AgsGstreamerDevout
    &#x251c;&#x2500;&#x2500; AgsGstreamerFile
    &#x251c;&#x2500;&#x2500; AgsGstreamerPort
    &#x251c;&#x2500;&#x2500; AgsGstreamerServer
    &#x251c;&#x2500;&#x2500; AgsIpatchDLS2Reader
    &#x251c;&#x2500;&#x2500; AgsIpatch
    &#x251c;&#x2500;&#x2500; AgsIpatchGigReader
    &#x251c;&#x2500;&#x2500; AgsIpatchSample
    &#x251c;&#x2500;&#x2500; AgsIpatchSF2Reader
    &#x251c;&#x2500;&#x2500; AgsJackClient
    &#x251c;&#x2500;&#x2500; AgsJackDevin
    &#x251c;&#x2500;&#x2500; AgsJackDevout
    &#x251c;&#x2500;&#x2500; AgsJackMidiin
```

```
    &#x251c;&#x2500;&#x2500; AgsJackPort
    &#x251c;&#x2500;&#x2500; AgsJackServer
    &#x251c;&#x2500;&#x2500; AgsConversion
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsLadspaConversion
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsLv2Conversion
    &#x251c;&#x2500;&#x2500; AgsLadspaManager
    &#x251c;&#x2500;&#x2500; AgsLv2EventManager
    &#x251c;&#x2500;&#x2500; AgsLv2LogManager
    &#x251c;&#x2500;&#x2500; AgsLv2Manager
    &#x251c;&#x2500;&#x2500; AgsLv2OptionManager
    &#x251c;&#x2500;&#x2500; AgsLv2Preset
    &#x251c;&#x2500;&#x2500; AgsLv2PresetManager
    &#x251c;&#x2500;&#x2500; AgsLv2TurtleParser
    &#x251c;&#x2500;&#x2500; AgsLv2TurtleScanner
    &#x251c;&#x2500;&#x2500; AgsLv2UriMapManager
    &#x251c;&#x2500;&#x2500; AgsLv2UridManager
    &#x251c;&#x2500;&#x2500; AgsLv2Worker
    &#x251c;&#x2500;&#x2500; AgsLv2WorkerManager
    &#x251c;&#x2500;&#x2500; AgsLv2uiManager
    &#x251c;&#x2500;&#x2500; AgsMidiBuilder
    &#x251c;&#x2500;&#x2500; AgsMidiFile
    &#x251c;&#x2500;&#x2500; AgsMidi
    &#x251c;&#x2500;&#x2500; AgsMidiParser
    &#x251c;&#x2500;&#x2500; AgsNotation
    &#x251c;&#x2500;&#x2500; AgsNote
    &#x251c;&#x2500;&#x2500; AgsOscController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscActionController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscConfigController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscExportController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscFrontController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscInfoController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscMeterController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscNodeController
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsOscRenewController
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsOscStatusController
    &#x251c;&#x2500;&#x2500; AgsOscBuilder
    &#x251c;&#x2500;&#x2500; AgsOscClient
    &#x251c;&#x2500;&#x2500; AgsOscConnection
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsOscWebsocketConnection
    &#x251c;&#x2500;&#x2500; AgsOscMessage
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsOscXmlrpcMessage
    &#x251c;&#x2500;&#x2500; AgsOscParser
    &#x251c;&#x2500;&#x2500; AgsOscResponse
    &#x251c;&#x2500;&#x2500; AgsOscServer
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsOscXmlrpcServer
    &#x251c;&#x2500;&#x2500; AgsController
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsOscXmlrpcController
    &#x251c;&#x2500;&#x2500; AgsOssDevin
    &#x251c;&#x2500;&#x2500; AgsOssDevout
    &#x251c;&#x2500;&#x2500; AgsOssMidiin
    &#x251c;&#x2500;&#x2500; AgsPattern
    &#x251c;&#x2500;&#x2500; AgsPlaybackDomain
    &#x251c;&#x2500;&#x2500; AgsPlayback
    &#x251c;&#x2500;&#x2500; AgsPluginPort
    &#x251c;&#x2500;&#x2500; AgsPort
    &#x251c;&#x2500;&#x2500; AgsPreset
    &#x251c;&#x2500;&#x2500; AgsPulseClient
    &#x251c;&#x2500;&#x2500; AgsPulseDevin
    &#x251c;&#x2500;&#x2500; AgsPulseDevout
    &#x251c;&#x2500;&#x2500; AgsPulsePort
    &#x251c;&#x2500;&#x2500; AgsPulseServer
    &#x251c;&#x2500;&#x2500; AgsRecallContainer
```

```
    &#x251c;&#x2500;&#x2500; AgsRecallDependency
    &#x251c;&#x2500;&#x2500; AgsRecallID
    &#x251c;&#x2500;&#x2500; AgsRecyclingContext
    &#x251c;&#x2500;&#x2500; AgsRecycling
    &#x251c;&#x2500;&#x2500; AgsSF2Loader
    &#x251c;&#x2500;&#x2500; AgsSF2MidiLocaleLoader
    &#x251c;&#x2500;&#x2500; AgsFunction
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsSF2SynthGenerator
    &#x2502;   &#x251c;&#x2500;&#x2500; AgsSFZSynthGenerator
    &#x2502;   &#x2570;&#x2500;&#x2500; AgsSynthGenerator
    &#x251c;&#x2500;&#x2500; AgsSFZFile
    &#x251c;&#x2500;&#x2500; AgsSFZGroup
    &#x251c;&#x2500;&#x2500; AgsSFZInstrumentLoader
    &#x251c;&#x2500;&#x2500; AgsSFZLoader
    &#x251c;&#x2500;&#x2500; AgsSFZRegion
    &#x251c;&#x2500;&#x2500; AgsSFZSample
    &#x251c;&#x2500;&#x2500; AgsSndfile
    &#x251c;&#x2500;&#x2500; AgsTrack
    &#x251c;&#x2500;&#x2500; AgsWasapiDevin
    &#x251c;&#x2500;&#x2500; AgsWasapiDevout
    &#x251c;&#x2500;&#x2500; AgsWave
    &#x2570;&#x2500;&#x2500; AgsWaveLoader
    GInterface
    &#x251c;&#x2500;&#x2500; AgsSoundProvider
    &#x251c;&#x2500;&#x2500; AgsSoundResource
    &#x251c;&#x2500;&#x2500; AgsSoundContainer
    &#x2570;&#x2500;&#x2500; AgsOscPluginController
    GFlags
    &#x251c;&#x2500;&#x2500; AgsAccelerationFlags
    &#x251c;&#x2500;&#x2500; AgsAlsaDevinFlags
    &#x251c;&#x2500;&#x2500; AgsAlsaDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsAudioFlags
    &#x251c;&#x2500;&#x2500; AgsAudioSignalFlags
    &#x251c;&#x2500;&#x2500; AgsAudioUnitClientFlags
    &#x251c;&#x2500;&#x2500; AgsAudioUnitDevinFlags
    &#x251c;&#x2500;&#x2500; AgsAudioUnitDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsAudioUnitPortFlags
    &#x251c;&#x2500;&#x2500; AgsAutomationFlags
    &#x251c;&#x2500;&#x2500; AgsBasePluginFlags
    &#x251c;&#x2500;&#x2500; AgsBufferFlags
    &#x251c;&#x2500;&#x2500; AgsChannelFlags
    &#x251c;&#x2500;&#x2500; AgsCoreAudioClientFlags
    &#x251c;&#x2500;&#x2500; AgsCoreAudioDevinFlags
    &#x251c;&#x2500;&#x2500; AgsCoreAudioDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsCoreAudioMidiinFlags
    &#x251c;&#x2500;&#x2500; AgsCoreAudioPortFlags
    &#x251c;&#x2500;&#x2500; AgsFifooutFlags
    &#x251c;&#x2500;&#x2500; AgsFluidIIRFilterType
    &#x251c;&#x2500;&#x2500; AgsFxFactoryCreateFlags
    &#x251c;&#x2500;&#x2500; AgsGstreamerClientFlags
    &#x251c;&#x2500;&#x2500; AgsGstreamerDevinFlags
    &#x251c;&#x2500;&#x2500; AgsGstreamerDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsGstreamerPortFlags
    &#x251c;&#x2500;&#x2500; AgsIpatchFlags
    &#x251c;&#x2500;&#x2500; AgsJackClientFlags
    &#x251c;&#x2500;&#x2500; AgsJackDevinFlags
    &#x251c;&#x2500;&#x2500; AgsJackDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsJackMidiinFlags
    &#x251c;&#x2500;&#x2500; AgsJackPortFlags
    &#x251c;&#x2500;&#x2500; AgsLadspaConversionFlags
    &#x251c;&#x2500;&#x2500; AgsLv2ConversionFlags
    &#x251c;&#x2500;&#x2500; AgsLv2PluginFlags
```

```
    &#x251c;&#x2500;&#x2500; AgsLv2TurtleScannerStatusFlags
    &#x251c;&#x2500;&#x2500; AgsLv2WorkerFlags
    &#x251c;&#x2500;&#x2500; AgsLv2uiPluginFlags
    &#x251c;&#x2500;&#x2500; AgsNotationFlags
    &#x251c;&#x2500;&#x2500; AgsOscClientFlags
    &#x251c;&#x2500;&#x2500; AgsOscFrontControllerFlags
    &#x251c;&#x2500;&#x2500; AgsOscServerFlags
    &#x251c;&#x2500;&#x2500; AgsOscXmlrpcControllerFlags
    &#x251c;&#x2500;&#x2500; AgsOssDevinFlags
    &#x251c;&#x2500;&#x2500; AgsOssDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsOssMidiinFlags
    &#x251c;&#x2500;&#x2500; AgsPlaybackFlags
    &#x251c;&#x2500;&#x2500; AgsPluginPortFlags
    &#x251c;&#x2500;&#x2500; AgsPortFlags
    &#x251c;&#x2500;&#x2500; AgsPulseClientFlags
    &#x251c;&#x2500;&#x2500; AgsPulseDevinFlags
    &#x251c;&#x2500;&#x2500; AgsPulseDevoutFlags
    &#x251c;&#x2500;&#x2500; AgsPulsePortFlags
    &#x251c;&#x2500;&#x2500; AgsRecallContainerFlags
    &#x251c;&#x2500;&#x2500; AgsRecallFlags
    &#x251c;&#x2500;&#x2500; AgsRecallNotifyDependencyMode
    &#x251c;&#x2500;&#x2500; AgsRecyclingContextFlags
    &#x251c;&#x2500;&#x2500; AgsRecyclingFlags
    &#x251c;&#x2500;&#x2500; AgsSndfileFlags
    &#x251c;&#x2500;&#x2500; AgsSoundAbilityFlags
    &#x251c;&#x2500;&#x2500; AgsSoundBehaviourFlags
    &#x251c;&#x2500;&#x2500; AgsSoundStagingFlags
    &#x251c;&#x2500;&#x2500; AgsSoundStateFlags
    &#x251c;&#x2500;&#x2500; AgsTrackFlags
    &#x251c;&#x2500;&#x2500; AgsWasapiDevinFlags
    &#x251c;&#x2500;&#x2500; AgsWasapiDevoutFlags
    &#x2570;&#x2500;&#x2500; AgsWaveFlags
    GBoxed
    &#x251c;&#x2500;&#x2500; AgsAudioBufferUtil
    &#x251c;&#x2500;&#x2500; AgsCharBufferUtil
    &#x251c;&#x2500;&#x2500; AgsChorusUtil
    &#x251c;&#x2500;&#x2500; AgsCommonPitchUtil
    &#x251c;&#x2500;&#x2500; AgsDiatonicScale
    &#x251c;&#x2500;&#x2500; AgsEnvelopeUtil
    &#x251c;&#x2500;&#x2500; AgsFastPitchUtil
    &#x251c;&#x2500;&#x2500; AgsFluidIIRFilterUtil
    &#x251c;&#x2500;&#x2500; AgsFluidInterpolate4thOrderUtil
    &#x251c;&#x2500;&#x2500; AgsFluidInterpolate7thOrderUtil
    &#x251c;&#x2500;&#x2500; AgsFluidInterpolateLinearUtil
    &#x251c;&#x2500;&#x2500; AgsFluidInterpolateNoneUtil
    &#x251c;&#x2500;&#x2500; AgsFluidUtil
    &#x251c;&#x2500;&#x2500; AgsFMSynthUtil
    &#x251c;&#x2500;&#x2500; AgsFourierTransformUtil
    &#x251c;&#x2500;&#x2500; AgsFrequencyAliaseUtil
    &#x251c;&#x2500;&#x2500; AgsHQPitchUtil
    &#x251c;&#x2500;&#x2500; AgsLFOSynthUtil
    &#x251c;&#x2500;&#x2500; AgsLinearInterpolateUtil
    &#x251c;&#x2500;&#x2500; AgsMidiBufferUtil
    &#x251c;&#x2500;&#x2500; AgsMidiUtil
    &#x251c;&#x2500;&#x2500; AgsNoiseUtil
    &#x251c;&#x2500;&#x2500; AgsOscBufferUtil
    &#x251c;&#x2500;&#x2500; AgsOscUtil
    &#x251c;&#x2500;&#x2500; AgsPeakUtil
    &#x251c;&#x2500;&#x2500; AgsPhaseShiftUtil
    &#x251c;&#x2500;&#x2500; AgsPortUtil
    &#x251c;&#x2500;&#x2500; AgsResampleUtil
    &#x251c;&#x2500;&#x2500; AgsSequencerUtil
```

```
    &#x251c;&#x2500;&#x2500; AgsSF2SynthUtil
    &#x251c;&#x2500;&#x2500; AgsSFZSynthUtil
    &#x251c;&#x2500;&#x2500; AgsSoundcardUtil
    &#x251c;&#x2500;&#x2500; AgsSynthUtil
    &#x251c;&#x2500;&#x2500; AgsTimeStretchUtil
    &#x2570;&#x2500;&#x2500; AgsVolumeUtil
    GEnum
    &#x251c;&#x2500;&#x2500; AgsSF2SynthUtilLoopMode
    &#x251c;&#x2500;&#x2500; AgsSFZSynthUtilLoopMode
    &#x251c;&#x2500;&#x2500; AgsSoundKeyFormat
    &#x251c;&#x2500;&#x2500; AgsSoundScope
    &#x2570;&#x2500;&#x2500; AgsSynthOscillatorMode
```

Chapter 25

Index

A
AGS_ACCELERATION, 1241
AgsAcceleration, 1243
AgsAcceleration:acceleration-name, 1244
AgsAcceleration:x, 1244
AgsAcceleration:y, 1244
AGS_ACCELERATION_CLASS, 1241
ags_acceleration_duplicate, 1241
ags_acceleration_flags_get_type, 1242
AGS_ACCELERATION_GET_CLASS, 1242
AGS_ACCELERATION_GET_OBJ_MUTEX, 1238
ags_acceleration_get_obj_mutex, 1238
ags_acceleration_get_type, 1242
ags_acceleration_get_x, 1239
ags_acceleration_get_y, 1240
ags_acceleration_new, 1241
ags_acceleration_set_flags, 1238
ags_acceleration_set_x, 1240
ags_acceleration_set_y, 1240
ags_acceleration_sort_func, 1239
ags_acceleration_test_flags, 1238
ags_acceleration_unset_flags, 1239
AgsAccelerationClass, 1243
AgsAccelerationFlags, 1242
AGS_ADD_AUDIO, 2271
AGS_ADD_AUDIO_CLASS, 2271
AGS_ADD_AUDIO_GET_CLASS, 2272
ags_add_audio_get_type, 2272
ags_add_audio_new, 2271
AGS_ADD_AUDIO_SIGNAL, 2274
AGS_ADD_AUDIO_SIGNAL_CLASS, 2274
AGS_ADD_AUDIO_SIGNAL_GET_CLASS, 2274
ags_add_audio_signal_get_type, 2275
ags_add_audio_signal_new, 2274
AGS_ADD_NOTE, 2278
AGS_ADD_NOTE_CLASS, 2278
AGS_ADD_NOTE_GET_CLASS, 2278
ags_add_note_get_type, 2278
ags_add_note_new, 2277
AGS_ADD_SOUNDCARD, 2281
AGS_ADD_SOUNDCARD_CLASS, 2281
AGS_ADD_SOUNDCARD_GET_CLASS, 2281
ags_add_soundcard_get_type, 2281
ags_add_soundcard_new, 2280
AgsAddAudio, 2272
AgsAddAudio:audio, 2272
AgsAddAudioClass, 2272
AgsAddAudioSignal, 2275
AgsAddAudioSignal:audio-signal, 2275
AgsAddAudioSignal:audio-signal-flags, 2275
AgsAddAudioSignal:recall-id, 2276
AgsAddAudioSignal:recycling, 2276
AgsAddAudioSignal:soundcard, 2276
AgsAddAudioSignalClass, 2275
AgsAddNote, 2278
AgsAddNote:audio, 2279
AgsAddNote:audio-channel, 2279
AgsAddNote:note, 2279
AgsAddNote:use-selection-list, 2279
AgsAddNoteClass, 2278
AgsAddSoundcard, 2282
AgsAddSoundcard:soundcard, 2282
AgsAddSoundcardClass, 2282
AGS_ALSA_DEVIN, 649
ags_alsa_devin_adjust_delay_and_attack, 648
AGS_ALSA_DEVIN_CLASS, 649
AGS_ALSA_DEVIN_DEFAULT_ALSA_DEVICE, 649
AGS_ALSA_DEVIN_DEFAULT_APP_BUFFER_SIZE, 649
AGS_ALSA_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE, 650
AGS_ALSA_DEVIN_ERROR, 651
ags_alsa_devin_error_quark, 646
ags_alsa_devin_flags_get_type, 649
AGS_ALSA_DEVIN_GET_CLASS, 649
AGS_ALSA_DEVIN_GET_OBJ_MUTEX, 646
ags_alsa_devin_get_type, 649
ags_alsa_devin_new, 648
ags_alsa_devin_realloc_buffer, 648
ags_alsa_devin_set_flags, 647
ags_alsa_devin_switch_buffer_flag, 648
ags_alsa_devin_test_flags, 647
ags_alsa_devin_unset_flags, 647
AGS_ALSA_DEVOUT, 658
ags_alsa_devout_adjust_delay_and_attack, 658
AGS_ALSA_DEVOUT_CLASS, 658
AGS_ALSA_DEVOUT_DEFAULT_ALSA_DEVICE, 659

AGS_ALSA_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 659
AGS_ALSA_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE, 659
AGS_ALSA_DEVOUT_ERROR, 661
ags_alsa_devout_error_quark, 656
ags_alsa_devout_flags_get_type, 659
AGS_ALSA_DEVOUT_GET_CLASS, 659
AGS_ALSA_DEVOUT_GET_OBJ_MUTEX, 656
ags_alsa_devout_get_type, 659
ags_alsa_devout_new, 658
ags_alsa_devout_realloc_buffer, 658
ags_alsa_devout_set_flags, 657
ags_alsa_devout_switch_buffer, 657
ags_alsa_devout_test_flags, 656
ags_alsa_devout_unset_flags, 657
AGS_ALSA_MIDIIN, 667
AGS_ALSA_MIDIIN_CLASS, 667
AGS_ALSA_MIDIIN_DEFAULT_ALSA_DEVICE, 668
AGS_ALSA_MIDIIN_DEFAULT_APP_BUFFER_SIZE, 668
AGS_ALSA_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE, 668
AGS_ALSA_MIDIIN_DEFAULT_BUFFER_SIZE, 668
AGS_ALSA_MIDIIN_ERROR, 671
ags_alsa_midiin_error_quark, 666
AGS_ALSA_MIDIIN_GET_CLASS, 668
AGS_ALSA_MIDIIN_GET_OBJ_MUTEX, 666
ags_alsa_midiin_get_type, 668
ags_alsa_midiin_new, 667
ags_alsa_midiin_set_flags, 666
ags_alsa_midiin_switch_buffer_flag, 667
ags_alsa_midiin_test_flags, 666
ags_alsa_midiin_unset_flags, 667
AgsAlsaDevin, 652
AgsAlsaDevin:attack, 652
AgsAlsaDevin:bpm, 652
AgsAlsaDevin:buffer, 653
AgsAlsaDevin:buffer-size, 653
AgsAlsaDevin:delay-factor, 653
AgsAlsaDevin:device, 653
AgsAlsaDevin:dsp-channels, 654
AgsAlsaDevin:format, 654
AgsAlsaDevin:pcm-channels, 654
AgsAlsaDevin:samplerate, 654
AgsAlsaDevinAppBufferMode, 650
AgsAlsaDevinBackendBufferMode, 651
AgsAlsaDevinClass, 652
AgsAlsaDevinError, 651
AgsAlsaDevinFlags, 650
AgsAlsaDevout, 662
AgsAlsaDevout:attack, 662
AgsAlsaDevout:bpm, 662
AgsAlsaDevout:buffer, 662
AgsAlsaDevout:buffer-size, 663
AgsAlsaDevout:delay-factor, 663
AgsAlsaDevout:device, 663
AgsAlsaDevout:dsp-channels, 663
AgsAlsaDevout:format, 664
AgsAlsaDevout:pcm-channels, 664
AgsAlsaDevout:samplerate, 664
AgsAlsaDevoutAppBufferMode, 660
AgsAlsaDevoutBackendBufferMode, 661
AgsAlsaDevoutClass, 662
AgsAlsaDevoutError, 661
AgsAlsaDevoutFlags, 659
AgsAlsaMidiin, 672
AgsAlsaMidiin:attack, 672
AgsAlsaMidiin:bpm, 672
AgsAlsaMidiin:buffer, 672
AgsAlsaMidiin:delay-factor, 673
AgsAlsaMidiin:device, 673
AgsAlsaMidiinAppBufferMode, 669
AgsAlsaMidiinBackendBufferMode, 670
AgsAlsaMidiinClass, 672
AgsAlsaMidiinError, 671
AgsAlsaMidiinFlags, 668
AgsAlsaMidiinSyncFlags, 670
AGS_APPLY_BPM, 2283
AGS_APPLY_BPM_CLASS, 2283
AGS_APPLY_BPM_GET_CLASS, 2284
ags_apply_bpm_get_type, 2284
ags_apply_bpm_new, 2283
AGS_APPLY_PRESETS, 2286
AGS_APPLY_PRESETS_CLASS, 2286
AGS_APPLY_PRESETS_GET_CLASS, 2287
ags_apply_presets_get_type, 2287
ags_apply_presets_new, 2286
ags_apply_presets_soundcard, 2286
AGS_APPLY_SEQUENCER_LENGTH, 2290
AGS_APPLY_SEQUENCER_LENGTH_CLASS, 2290
AGS_APPLY_SEQUENCER_LENGTH_GET_CLASS, 2290
ags_apply_sequencer_length_get_type, 2290
ags_apply_sequencer_length_new, 2289
AGS_APPLY_SF2_MIDI_LOCALE, 2303
AGS_APPLY_SF2_MIDI_LOCALE_CLASS, 2303
AGS_APPLY_SF2_MIDI_LOCALE_GET_CLASS, 2303
ags_apply_sf2_midi_locale_get_type, 2303
ags_apply_sf2_midi_locale_new, 2302
AGS_APPLY_SF2_SYNTH, 2299
AGS_APPLY_SF2_SYNTH_CLASS, 2299
AGS_APPLY_SF2_SYNTH_GET_CLASS, 2299
ags_apply_sf2_synth_get_type, 2300
ags_apply_sf2_synth_new, 2299
AGS_APPLY_SFZ_INSTRUMENT, 2309
AGS_APPLY_SFZ_INSTRUMENT_CLASS, 2309
AGS_APPLY_SFZ_INSTRUMENT_GET_CLASS, 2309
ags_apply_sfz_instrument_get_type, 2310
ags_apply_sfz_instrument_new, 2309
AGS_APPLY_SFZ_SYNTH, 2306
AGS_APPLY_SFZ_SYNTH_CLASS, 2306
AGS_APPLY_SFZ_SYNTH_GET_CLASS, 2306
ags_apply_sfz_synth_get_type, 2306
ags_apply_sfz_synth_new, 2305
AGS_APPLY_SOUND_CONFIG, 2293

AGS_APPLY_SOUND_CONFIG_CLASS, 2293
AGS_APPLY_SOUND_CONFIG_GET_CLASS, 2293
ags_apply_sound_config_get_type, 2293
ags_apply_sound_config_new, 2292
AGS_APPLY_SYNTH, 2295
AGS_APPLY_SYNTH_CLASS, 2296
AGS_APPLY_SYNTH_GET_CLASS, 2296
ags_apply_synth_get_type, 2296
ags_apply_synth_new, 2295
AGS_APPLY_TACT, 2312
AGS_APPLY_TACT_CLASS, 2312
AGS_APPLY_TACT_GET_CLASS, 2312
ags_apply_tact_get_type, 2312
ags_apply_tact_new, 2311
AgsApplyBpm, 2284
AgsApplyBpm:bpm, 2284
AgsApplyBpm:scope, 2285
AgsApplyBpmClass, 2284
AgsApplyPresets, 2287
AgsApplyPresets:buffer-size, 2287
AgsApplyPresets:format, 2288
AgsApplyPresets:pcm-channels, 2288
AgsApplyPresets:samplerate, 2288
AgsApplyPresets:scope, 2288
AgsApplyPresetsClass, 2287
AgsApplySequencerLength, 2291
AgsApplySequencerLength:scope, 2291
AgsApplySequencerLength:sequencer-length, 2291
AgsApplySequencerLengthClass, 2291
AgsApplySF2MidiLocale, 2304
AgsApplySF2MidiLocale:synth, 2304
AgsApplySF2MidiLocale:template, 2304
AgsApplySF2MidiLocaleClass, 2304
AgsApplySF2Synth, 2300
AgsApplySF2Synth:base-note, 2300
AgsApplySF2Synth:count, 2301
AgsApplySF2Synth:requested-frame-count, 2301
AgsApplySF2Synth:sf2-synth-generator, 2301
AgsApplySF2Synth:start-channel, 2301
AgsApplySF2SynthClass, 2300
AgsApplySFZInstrument, 2310
AgsApplySFZInstrument:synth, 2310
AgsApplySFZInstrument:template, 2310
AgsApplySFZInstrumentClass, 2310
AgsApplySFZSynth, 2306
AgsApplySFZSynth:base-note, 2307
AgsApplySFZSynth:count, 2307
AgsApplySFZSynth:requested-frame-count, 2307
AgsApplySFZSynth:sfz-synth-generator, 2307
AgsApplySFZSynth:start-channel, 2308
AgsApplySFZSynthClass, 2307
AgsApplySoundConfig, 2293
AgsApplySoundConfig:config-data, 2294
AgsApplySoundConfigClass, 2294
AgsApplySynth, 2296
AgsApplySynth:base-note, 2297
AgsApplySynth:count, 2297
AgsApplySynth:requested-frame-count, 2297
AgsApplySynth:start-channel, 2297
AgsApplySynth:synth-generator, 2298
AgsApplySynthClass, 2296
AgsApplyTact, 2313
AgsApplyTact:scope, 2313
AgsApplyTact:tact, 2313
AgsApplyTactClass, 2313
AGS_AUDIO, 1020
AgsAudio, 1022
AgsAudio::cancel-recall, 1036
AgsAudio::check-scope, 1036
AgsAudio::cleanup-recall, 1036
AgsAudio::done-recall, 1037
AgsAudio::duplicate-recall, 1037
AgsAudio::init-recall, 1038
AgsAudio::play-recall, 1038
AgsAudio::recursive-run-stage, 1039
AgsAudio::resolve-recall, 1039
AgsAudio::set-audio-channels, 1039
AgsAudio::set-pads, 1040
AgsAudio::start, 1040
AgsAudio::stop, 1041
AgsAudio:absolute-key, 1023
AgsAudio:audio-channels, 1023
AgsAudio:audio-end-mapping, 1023
AgsAudio:audio-name, 1024
AgsAudio:audio-start-mapping, 1024
AgsAudio:automation, 1024
AgsAudio:bpm, 1024
AgsAudio:buffer-size, 1025
AgsAudio:cursor, 1025
AgsAudio:denominator, 1025
AgsAudio:format, 1025
AgsAudio:input, 1026
AgsAudio:input-audio-file, 1026
AgsAudio:input-lines, 1026
AgsAudio:input-midi-file, 1026
AgsAudio:input-pads, 1026
AgsAudio:input-sequencer, 1027
AgsAudio:input-soundcard, 1027
AgsAudio:is-minor, 1027
AgsAudio:key, 1027
AgsAudio:loop-end, 1027
AgsAudio:loop-start, 1028
AgsAudio:max-audio-channels, 1028
AgsAudio:max-input-pads, 1028
AgsAudio:max-output-pads, 1028
AgsAudio:midi, 1028
AgsAudio:midi-channel, 1029
AgsAudio:midi-end-mapping, 1029
AgsAudio:midi-start-mapping, 1029
AgsAudio:min-audio-channels, 1029
AgsAudio:min-input-pads, 1030
AgsAudio:min-output-pads, 1030
AgsAudio:notation, 1030
AgsAudio:numerator, 1030

AgsAudio:octave, 1031
AgsAudio:offset, 1031
AgsAudio:output, 1031
AgsAudio:output-audio-file, 1031
AgsAudio:output-lines, 1031
AgsAudio:output-midi-file, 1032
AgsAudio:output-pads, 1032
AgsAudio:output-sequencer, 1032
AgsAudio:output-soundcard, 1032
AgsAudio:play, 1032
AgsAudio:playback-domain, 1033
AgsAudio:preset, 1033
AgsAudio:recall, 1033
AgsAudio:recall-container, 1033
AgsAudio:recall-id, 1033
AgsAudio:recycling-context, 1034
AgsAudio:samplerate, 1034
AgsAudio:sf2-synth-generator, 1034
AgsAudio:sfz-synth-generator, 1034
AgsAudio:sharp-flats, 1035
AgsAudio:synth-generator, 1035
AgsAudio:time-signature, 1035
AgsAudio:wave, 1035
ags_audio_add_automation, 1000
ags_audio_add_automation_port, 999
ags_audio_add_cursor, 996
ags_audio_add_midi, 1005
ags_audio_add_notation, 997
ags_audio_add_preset, 990
ags_audio_add_recall, 1012
ags_audio_add_recall_container, 1010
ags_audio_add_recall_id, 1008
ags_audio_add_recycling_context, 1009
ags_audio_add_sf2_synth_generator, 993
ags_audio_add_sfz_synth_generator, 995
ags_audio_add_synth_generator, 992
ags_audio_add_wave, 1001
AGS_AUDIO_APPLICATION_CONTEXT, 956
AGS_AUDIO_APPLICATION_CONTEXT_CLASS, 957
AGS_AUDIO_APPLICATION_CONTEXT_DEFAULT_LOADER_INTERVAL, 957
AGS_AUDIO_APPLICATION_CONTEXT_GET_CLASS, 957
ags_audio_application_context_get_type, 957
ags_audio_application_context_loader_timeout, 956
ags_audio_application_context_new, 956
AGS_AUDIO_BUFFER_COMPLEX, 270
AGS_AUDIO_BUFFER_DOUBLE, 270
AGS_AUDIO_BUFFER_FLOAT, 270
AGS_AUDIO_BUFFER_S16, 270
AGS_AUDIO_BUFFER_S24, 270
AGS_AUDIO_BUFFER_S32, 270
AGS_AUDIO_BUFFER_S64, 270
AGS_AUDIO_BUFFER_S8, 269
ags_audio_buffer_util_clear_buffer, 274
ags_audio_buffer_util_clear_complex, 273
ags_audio_buffer_util_clear_double, 273
ags_audio_buffer_util_clear_float, 272
ags_audio_buffer_util_clear_float32, 273
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_DOUBLE, 271
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_FLOAT, 271
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S16, 271
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S24, 271
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S32, 271
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S64, 271
AGS_AUDIO_BUFFER_UTIL_COMPLEX_TO_S8, 271
ags_audio_buffer_util_copy_buffer_to_buffer, 314
ags_audio_buffer_util_copy_complex_to_complex, 310
ags_audio_buffer_util_copy_complex_to_double, 313
ags_audio_buffer_util_copy_complex_to_float, 312
ags_audio_buffer_util_copy_complex_to_float32, 313
ags_audio_buffer_util_copy_complex_to_s16, 311
ags_audio_buffer_util_copy_complex_to_s24, 311
ags_audio_buffer_util_copy_complex_to_s32, 311
ags_audio_buffer_util_copy_complex_to_s64, 312
ags_audio_buffer_util_copy_complex_to_s8, 310
ags_audio_buffer_util_copy_double_to_complex, 309
ags_audio_buffer_util_copy_double_to_double, 299
ags_audio_buffer_util_copy_double_to_float, 298
ags_audio_buffer_util_copy_double_to_float32, 302
ags_audio_buffer_util_copy_double_to_s16, 296
ags_audio_buffer_util_copy_double_to_s24, 297
ags_audio_buffer_util_copy_double_to_s32, 297
ags_audio_buffer_util_copy_double_to_s64, 298
ags_audio_buffer_util_copy_double_to_s8, 296
ags_audio_buffer_util_copy_float32_to_complex, 309
ags_audio_buffer_util_copy_float32_to_double, 305
ags_audio_buffer_util_copy_float32_to_float, 305
ags_audio_buffer_util_copy_float32_to_float32, 302
ags_audio_buffer_util_copy_float32_to_s16, 303
ags_audio_buffer_util_copy_float32_to_s24, 304
ags_audio_buffer_util_copy_float32_to_s32, 304
ags_audio_buffer_util_copy_float32_to_s64, 305
ags_audio_buffer_util_copy_float32_to_s8, 303
ags_audio_buffer_util_copy_float_to_complex, 308
ags_audio_buffer_util_copy_float_to_double, 295
ags_audio_buffer_util_copy_float_to_float, 295
ags_audio_buffer_util_copy_float_to_float32, 301
ags_audio_buffer_util_copy_float_to_s16, 293
ags_audio_buffer_util_copy_float_to_s24, 293
ags_audio_buffer_util_copy_float_to_s32, 294
ags_audio_buffer_util_copy_float_to_s64, 294
ags_audio_buffer_util_copy_float_to_s8, 293
ags_audio_buffer_util_copy_s16_to_complex, 306
ags_audio_buffer_util_copy_s16_to_double, 283
ags_audio_buffer_util_copy_s16_to_float, 283
ags_audio_buffer_util_copy_s16_to_float32, 299
ags_audio_buffer_util_copy_s16_to_s16, 281
ags_audio_buffer_util_copy_s16_to_s24, 281
ags_audio_buffer_util_copy_s16_to_s32, 282
ags_audio_buffer_util_copy_s16_to_s64, 282
ags_audio_buffer_util_copy_s16_to_s8, 281
ags_audio_buffer_util_copy_s24_to_complex, 307

ags_audio_buffer_util_copy_s24_to_double, 286
ags_audio_buffer_util_copy_s24_to_float, 286
ags_audio_buffer_util_copy_s24_to_float32, 300
ags_audio_buffer_util_copy_s24_to_s16, 284
ags_audio_buffer_util_copy_s24_to_s24, 284
ags_audio_buffer_util_copy_s24_to_s32, 285
ags_audio_buffer_util_copy_s24_to_s64, 285
ags_audio_buffer_util_copy_s24_to_s8, 284
ags_audio_buffer_util_copy_s32_to_complex, 307
ags_audio_buffer_util_copy_s32_to_double, 289
ags_audio_buffer_util_copy_s32_to_float, 289
ags_audio_buffer_util_copy_s32_to_float32, 300
ags_audio_buffer_util_copy_s32_to_s16, 287
ags_audio_buffer_util_copy_s32_to_s24, 287
ags_audio_buffer_util_copy_s32_to_s32, 288
ags_audio_buffer_util_copy_s32_to_s64, 288
ags_audio_buffer_util_copy_s32_to_s8, 287
ags_audio_buffer_util_copy_s64_to_complex, 308
ags_audio_buffer_util_copy_s64_to_double, 292
ags_audio_buffer_util_copy_s64_to_float, 292
ags_audio_buffer_util_copy_s64_to_float32, 301
ags_audio_buffer_util_copy_s64_to_s16, 290
ags_audio_buffer_util_copy_s64_to_s24, 290
ags_audio_buffer_util_copy_s64_to_s32, 291
ags_audio_buffer_util_copy_s64_to_s64, 291
ags_audio_buffer_util_copy_s64_to_s8, 290
ags_audio_buffer_util_copy_s8_to_complex, 306
ags_audio_buffer_util_copy_s8_to_double, 280
ags_audio_buffer_util_copy_s8_to_float, 280
ags_audio_buffer_util_copy_s8_to_float32, 299
ags_audio_buffer_util_copy_s8_to_s16, 278
ags_audio_buffer_util_copy_s8_to_s24, 278
ags_audio_buffer_util_copy_s8_to_s32, 279
ags_audio_buffer_util_copy_s8_to_s64, 279
ags_audio_buffer_util_copy_s8_to_s8, 278
AGS_AUDIO_BUFFER_UTIL_DOUBLE_TO_COMPLEX, 271
AGS_AUDIO_BUFFER_UTIL_FLOAT_TO_COMPLEX, 271
ags_audio_buffer_util_format_from_soundcard, 272
ags_audio_buffer_util_get_copy_mode, 272
ags_audio_buffer_util_get_type, 314
ags_audio_buffer_util_pong, 277
ags_audio_buffer_util_pong_complex, 277
ags_audio_buffer_util_pong_double, 276
ags_audio_buffer_util_pong_float, 276
ags_audio_buffer_util_pong_s16, 274
ags_audio_buffer_util_pong_s24, 275
ags_audio_buffer_util_pong_s32, 275
ags_audio_buffer_util_pong_s64, 276
ags_audio_buffer_util_pong_s8, 274
AGS_AUDIO_BUFFER_UTIL_S16_TO_COMPLEX, 270
AGS_AUDIO_BUFFER_UTIL_S24_TO_COMPLEX, 270
AGS_AUDIO_BUFFER_UTIL_S32_TO_COMPLEX, 270
AGS_AUDIO_BUFFER_UTIL_S64_TO_COMPLEX, 271
AGS_AUDIO_BUFFER_UTIL_S8_TO_COMPLEX, 270
AGS_AUDIO_BUILD_ID, 957
ags_audio_cancel_recall, 1015
ags_audio_check_scope, 1016
AGS_AUDIO_CLASS, 1020
ags_audio_cleanup_recall, 1015
ags_audio_collect_all_audio_ports, 1017
ags_audio_collect_all_audio_ports_by_specifier_and_context, 1017
AGS_AUDIO_CONTAINER, 1789
ags_audio_container_add_audio_signal, 1785
ags_audio_container_add_wave, 1785
ags_audio_container_check_suffix, 1784
AGS_AUDIO_CONTAINER_CLASS, 1789
ags_audio_container_close, 1786
ags_audio_container_find_sound_resource, 1784
ags_audio_container_flush, 1788
AGS_AUDIO_CONTAINER_GET_CLASS, 1789
AGS_AUDIO_CONTAINER_GET_OBJ_MUTEX, 1783
ags_audio_container_get_type, 1789
AGS_AUDIO_CONTAINER_MANAGER, 1780
ags_audio_container_manager_add_audio_container, 1778
AGS_AUDIO_CONTAINER_MANAGER_CLASS, 1780
ags_audio_container_manager_find_audio_container, 1779
AGS_AUDIO_CONTAINER_MANAGER_GET_CLASS, 1780
ags_audio_container_manager_get_instance, 1779
AGS_AUDIO_CONTAINER_MANAGER_GET_OBJ_MUTEX, 1778
ags_audio_container_manager_get_obj_mutex, 1778
ags_audio_container_manager_get_type, 1780
ags_audio_container_manager_new, 1780
ags_audio_container_manager_remove_audio_container, 1779
ags_audio_container_new, 1788
ags_audio_container_open, 1786
ags_audio_container_open_from_data, 1786
ags_audio_container_read, 1787
ags_audio_container_read_audio_signal, 1787
ags_audio_container_read_wave, 1787
ags_audio_container_remove_audio_signal, 1785
ags_audio_container_remove_wave, 1785
ags_audio_container_rw_open, 1786
ags_audio_container_seek, 1788
ags_audio_container_set_flags, 1783
ags_audio_container_test_flags, 1783
ags_audio_container_unset_flags, 1783
ags_audio_container_write, 1788
AGS_AUDIO_DEFAULT_VERSION, 957
ags_audio_done_recall, 1015
ags_audio_duplicate_recall, 1013
AGS_AUDIO_FILE, 1811
ags_audio_file_add_audio_signal, 1807
ags_audio_file_add_wave, 1807
ags_audio_file_check_suffix, 1800
AGS_AUDIO_FILE_CLASS, 1812
ags_audio_file_close, 1809
ags_audio_file_flush, 1811
ags_audio_file_get_audio_channel, 1805
AGS_AUDIO_FILE_GET_CLASS, 1812
ags_audio_file_get_file_audio_channels, 1801
ags_audio_file_get_file_frame_count, 1803

ags_audio_file_get_file_samplerate, 1802
ags_audio_file_get_filename, 1801
ags_audio_file_get_format, 1805
ags_audio_file_get_frame_count, 1804
AGS_AUDIO_FILE_GET_OBJ_MUTEX, 1799
ags_audio_file_get_samplerate, 1803
ags_audio_file_get_sound_resource, 1806
ags_audio_file_get_soundcard, 1800
ags_audio_file_get_type, 1812
AGS_AUDIO_FILE_LINK, 1816
AGS_AUDIO_FILE_LINK_CLASS, 1816
AGS_AUDIO_FILE_LINK_GET_CLASS, 1817
ags_audio_file_link_get_type, 1817
ags_audio_file_link_new, 1816
AGS_AUDIO_FILE_MANAGER, 1795
ags_audio_file_manager_add_audio_file, 1794
AGS_AUDIO_FILE_MANAGER_CLASS, 1796
ags_audio_file_manager_find_audio_file, 1795
AGS_AUDIO_FILE_MANAGER_GET_CLASS, 1796
ags_audio_file_manager_get_instance, 1795
AGS_AUDIO_FILE_MANAGER_GET_OBJ_MUTEX, 1794
ags_audio_file_manager_get_obj_mutex, 1794
ags_audio_file_manager_get_type, 1796
ags_audio_file_manager_new, 1795
ags_audio_file_manager_remove_audio_file, 1794
ags_audio_file_new, 1811
ags_audio_file_open, 1808
ags_audio_file_open_from_data, 1808
ags_audio_file_read, 1809
ags_audio_file_read_audio_signal, 1809
ags_audio_file_read_wave, 1810
ags_audio_file_remove_audio_signal, 1807
ags_audio_file_remove_wave, 1807
ags_audio_file_rw_open, 1808
ags_audio_file_seek, 1810
ags_audio_file_set_audio_channel, 1806
ags_audio_file_set_file_audio_channels, 1802
ags_audio_file_set_file_frame_count, 1803
ags_audio_file_set_file_samplerate, 1802
ags_audio_file_set_filename, 1801
ags_audio_file_set_flags, 1799
ags_audio_file_set_format, 1805
ags_audio_file_set_frame_count, 1804
ags_audio_file_set_samplerate, 1804
ags_audio_file_set_sound_resource, 1806
ags_audio_file_set_soundcard, 1800
ags_audio_file_test_flags, 1799
ags_audio_file_unset_flags, 1799
ags_audio_file_write, 1810
ags_audio_find_name, 971
ags_audio_flags_get_type, 1020
ags_audio_get_absolute_key, 989
ags_audio_get_audio_channels, 974
ags_audio_get_audio_name, 970
ags_audio_get_audio_start_mapping, 983
ags_audio_get_automation, 999
ags_audio_get_automation_port, 998
ags_audio_get_bpm, 982
ags_audio_get_buffer_size, 981
AGS_AUDIO_GET_CLASS, 1020
ags_audio_get_cursor, 995
ags_audio_get_denominator, 985
ags_audio_get_format, 981
ags_audio_get_input, 977
ags_audio_get_input_audio_file, 1003
ags_audio_get_input_lines, 976
ags_audio_get_input_midi_file, 1006
ags_audio_get_input_pads, 975
ags_audio_get_input_sequencer, 979
ags_audio_get_input_soundcard, 978
ags_audio_get_instrument_name, 1003
ags_audio_get_is_minor, 986
ags_audio_get_key, 988
ags_audio_get_max_audio_channels, 971
ags_audio_get_max_input_pads, 973
ags_audio_get_max_output_pads, 972
ags_audio_get_midi, 1004
ags_audio_get_midi_channel, 984
ags_audio_get_midi_start_mapping, 983
ags_audio_get_notation, 997
ags_audio_get_numerator, 984
AGS_AUDIO_GET_OBJ_MUTEX, 964
ags_audio_get_obj_mutex, 964
ags_audio_get_octave, 988
ags_audio_get_output, 977
ags_audio_get_output_audio_file, 1002
ags_audio_get_output_lines, 976
ags_audio_get_output_midi_file, 1006
ags_audio_get_output_pads, 974
ags_audio_get_output_sequencer, 979
ags_audio_get_output_soundcard, 977
ags_audio_get_play, 1011
AGS_AUDIO_GET_PLAY_MUTEX, 964
ags_audio_get_play_mutex, 965
ags_audio_get_playback_domain, 991
ags_audio_get_preset, 989
ags_audio_get_recall, 1012
ags_audio_get_recall_container, 1010
ags_audio_get_recall_id, 1007
AGS_AUDIO_GET_RECALL_MUTEX, 964
ags_audio_get_recall_mutex, 965
ags_audio_get_recycling_context, 1008
ags_audio_get_samplerate, 980
ags_audio_get_sf2_synth_generator, 993
ags_audio_get_sfz_synth_generator, 994
ags_audio_get_sharp_flats, 987
ags_audio_get_synth_generator, 991
ags_audio_get_time_signature, 986
ags_audio_get_track_name, 1004
ags_audio_get_type, 1020
ags_audio_get_wave, 1001
ags_audio_init_recall, 1014
ags_audio_insert_recall, 1013
AGS_AUDIO_LOOP, 1454

ags_audio_loop_add_audio, 1451
ags_audio_loop_add_channel, 1452
AGS_AUDIO_LOOP_CLASS, 1454
AGS_AUDIO_LOOP_DEFAULT_JIFFIE, 1455
AGS_AUDIO_LOOP_GET_CLASS, 1454
ags_audio_loop_get_do_fx_staging, 1452
ags_audio_loop_get_staging_program, 1453
ags_audio_loop_get_type, 1455
ags_audio_loop_new, 1454
ags_audio_loop_remove_audio, 1451
ags_audio_loop_remove_channel, 1452
ags_audio_loop_set_do_fx_staging, 1453
ags_audio_loop_set_flags, 1451
ags_audio_loop_set_staging_program, 1453
ags_audio_loop_test_flags, 1450
ags_audio_loop_unset_flags, 1451
ags_audio_new, 1019
ags_audio_open_audio_file_as_channel, 1018
ags_audio_open_audio_file_as_wave, 1018
ags_audio_open_midi_file_as_midi, 1018
ags_audio_open_midi_file_as_notation, 1018
ags_audio_play_recall, 1014
ags_audio_recursive_run_stage, 1019
ags_audio_recursive_set_property, 1019
ags_audio_remove_automation, 1000
ags_audio_remove_automation_port, 999
ags_audio_remove_cursor, 996
ags_audio_remove_midi, 1005
ags_audio_remove_notation, 998
ags_audio_remove_preset, 990
ags_audio_remove_recall, 1013
ags_audio_remove_recall_container, 1011
ags_audio_remove_recall_id, 1008
ags_audio_remove_recycling_context, 1009
ags_audio_remove_sf2_synth_generator, 994
ags_audio_remove_sfz_synth_generator, 995
ags_audio_remove_synth_generator, 992
ags_audio_remove_wave, 1002
ags_audio_resolve_recall, 1014
ags_audio_set_ability_flags, 967
ags_audio_set_absolute_key, 989
ags_audio_set_audio_channels, 974
ags_audio_set_audio_name, 971
ags_audio_set_audio_start_mapping, 983
ags_audio_set_automation, 1000
ags_audio_set_automation_port, 998
ags_audio_set_behaviour_flags, 968
ags_audio_set_bpm, 982
ags_audio_set_buffer_size, 981
ags_audio_set_cursor, 996
ags_audio_set_denominator, 985
ags_audio_set_flags, 966
ags_audio_set_format, 982
ags_audio_set_input_audio_file, 1003
ags_audio_set_input_midi_file, 1007
ags_audio_set_input_pads, 975
ags_audio_set_input_sequencer, 980
ags_audio_set_input_soundcard, 978
ags_audio_set_instrument_name, 1004
ags_audio_set_is_minor, 987
ags_audio_set_key, 989
ags_audio_set_max_audio_channels, 972
ags_audio_set_max_input_pads, 973
ags_audio_set_max_output_pads, 972
ags_audio_set_max_pads, 973
ags_audio_set_midi, 1005
ags_audio_set_midi_channel, 984
ags_audio_set_midi_start_mapping, 984
ags_audio_set_notation, 997
ags_audio_set_numerator, 985
ags_audio_set_octave, 988
ags_audio_set_output_audio_file, 1002
ags_audio_set_output_midi_file, 1006
ags_audio_set_output_pads, 975
ags_audio_set_output_sequencer, 979
ags_audio_set_output_soundcard, 978
ags_audio_set_pads, 976
ags_audio_set_play, 1011
ags_audio_set_playback_domain, 991
ags_audio_set_preset, 990
ags_audio_set_recall, 1012
ags_audio_set_recall_container, 1010
ags_audio_set_recall_id, 1007
ags_audio_set_recycling_context, 1009
ags_audio_set_samplerate, 980
ags_audio_set_sf2_synth_generator, 993
ags_audio_set_sfz_synth_generator, 994
ags_audio_set_sharp_flats, 987
ags_audio_set_staging_completed, 970
ags_audio_set_staging_flags, 969
ags_audio_set_synth_generator, 992
ags_audio_set_time_signature, 986
ags_audio_set_track_name, 1004
ags_audio_set_wave, 1001
AGS_AUDIO_SIGNAL, 1141
ags_audio_signal_add_note, 1132
ags_audio_signal_add_stream, 1134
AGS_AUDIO_SIGNAL_CLASS, 1141
ags_audio_signal_clear, 1135
ags_audio_signal_close_feed, 1137
ags_audio_signal_contains_note, 1138
ags_audio_signal_continue_feed, 1137
ags_audio_signal_duplicate_stream, 1135
ags_audio_signal_feed, 1135
ags_audio_signal_feed_extended, 1136
ags_audio_signal_find_by_recall_id, 1139
ags_audio_signal_find_stream_current, 1139
ags_audio_signal_flags_get_type, 1141
ags_audio_signal_get_buffer_size, 1130
AGS_AUDIO_SIGNAL_GET_CLASS, 1141
ags_audio_signal_get_format, 1131
ags_audio_signal_get_input_soundcard, 1129
ags_audio_signal_get_length_till_current, 1137
ags_audio_signal_get_note, 1132

AGS_AUDIO_SIGNAL_GET_OBJ_MUTEX, 1124
ags_audio_signal_get_obj_mutex, 1124
ags_audio_signal_get_output_soundcard, 1128
ags_audio_signal_get_recycling, 1128
ags_audio_signal_get_rt_template, 1138
ags_audio_signal_get_samplerate, 1130
ags_audio_signal_get_stream, 1133
AGS_AUDIO_SIGNAL_GET_STREAM_MUTEX, 1124
ags_audio_signal_get_template, 1138
ags_audio_signal_get_type, 1142
ags_audio_signal_is_active, 1140
ags_audio_signal_new, 1140
ags_audio_signal_new_with_length, 1140
ags_audio_signal_open_feed, 1136
ags_audio_signal_refresh_data, 1134
ags_audio_signal_remove_note, 1133
ags_audio_signal_set_buffer_size, 1131
ags_audio_signal_set_flags, 1126
ags_audio_signal_set_format, 1131
ags_audio_signal_set_input_soundcard, 1129
ags_audio_signal_set_note, 1132
ags_audio_signal_set_output_soundcard, 1129
ags_audio_signal_set_recycling, 1128
ags_audio_signal_set_samplerate, 1130
ags_audio_signal_set_stream, 1133
ags_audio_signal_stream_lock, 1125
ags_audio_signal_stream_resize, 1134
ags_audio_signal_stream_safe_resize, 1134
ags_audio_signal_stream_unlock, 1125
ags_audio_signal_test_flags, 1125
ags_audio_signal_unset_flags, 1126
ags_audio_start, 1016
ags_audio_stop, 1016
ags_audio_test_ability_flags, 966
ags_audio_test_behaviour_flags, 967
ags_audio_test_flags, 965
ags_audio_test_staging_completed, 969
ags_audio_test_staging_flags, 968
AGS_AUDIO_THREAD, 1462
AGS_AUDIO_THREAD_CLASS, 1462
AGS_AUDIO_THREAD_DEFAULT_JIFFIE, 1463
AGS_AUDIO_THREAD_GET_CLASS, 1462
ags_audio_thread_get_do_fx_staging, 1460
ags_audio_thread_get_processing, 1459
ags_audio_thread_get_staging_program, 1461
ags_audio_thread_get_task_launcher, 1459
ags_audio_thread_get_type, 1463
ags_audio_thread_new, 1462
ags_audio_thread_scope_data_alloc, 1460
ags_audio_thread_scope_data_free, 1460
ags_audio_thread_set_do_fx_staging, 1461
ags_audio_thread_set_processing, 1459
ags_audio_thread_set_sound_scope, 1460
ags_audio_thread_set_staging_program, 1461
ags_audio_thread_set_status_flags, 1458
ags_audio_thread_test_status_flags, 1458
ags_audio_thread_unset_status_flags, 1458
AGS_AUDIO_UNIT_CLIENT, 918
ags_audio_unit_client_activate, 917
ags_audio_unit_client_add_device, 916
ags_audio_unit_client_add_port, 916
AGS_AUDIO_UNIT_CLIENT_CLASS, 918
ags_audio_unit_client_deactivate, 917
ags_audio_unit_client_find, 915
ags_audio_unit_client_find_uuid, 915
ags_audio_unit_client_flags_get_type, 918
AGS_AUDIO_UNIT_CLIENT_GET_CLASS, 918
AGS_AUDIO_UNIT_CLIENT_GET_OBJ_MUTEX, 914
ags_audio_unit_client_get_type, 919
ags_audio_unit_client_new, 918
ags_audio_unit_client_open, 916
ags_audio_unit_client_remove_device, 916
ags_audio_unit_client_remove_port, 917
ags_audio_unit_client_set_flags, 914
ags_audio_unit_client_test_flags, 914
ags_audio_unit_client_unset_flags, 914
AGS_AUDIO_UNIT_DEVIN, 931
ags_audio_unit_devin_adjust_delay_and_attack, 930
AGS_AUDIO_UNIT_DEVIN_CLASS, 931
AGS_AUDIO_UNIT_DEVIN_DEFAULT_APP_BUFFER_SIZE, 931
AGS_AUDIO_UNIT_DEVIN_ERROR, 934
ags_audio_unit_devin_error_quark, 928
ags_audio_unit_devin_flags_get_type, 931
AGS_AUDIO_UNIT_DEVIN_GET_CLASS, 931
AGS_AUDIO_UNIT_DEVIN_GET_OBJ_MUTEX, 928
ags_audio_unit_devin_get_type, 931
ags_audio_unit_devin_new, 930
ags_audio_unit_devin_realloc_buffer, 930
ags_audio_unit_devin_set_flags, 929
ags_audio_unit_devin_switch_buffer_flag, 930
ags_audio_unit_devin_test_flags, 929
ags_audio_unit_devin_unset_flags, 929
AGS_AUDIO_UNIT_DEVOUT, 941
ags_audio_unit_devout_adjust_delay_and_attack, 940
AGS_AUDIO_UNIT_DEVOUT_CLASS, 941
AGS_AUDIO_UNIT_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 942
AGS_AUDIO_UNIT_DEVOUT_ERROR, 944
ags_audio_unit_devout_error_quark, 939
ags_audio_unit_devout_flags_get_type, 941
AGS_AUDIO_UNIT_DEVOUT_GET_CLASS, 941
AGS_AUDIO_UNIT_DEVOUT_GET_OBJ_MUTEX, 939
ags_audio_unit_devout_get_type, 942
ags_audio_unit_devout_new, 941
ags_audio_unit_devout_realloc_buffer, 940
ags_audio_unit_devout_set_flags, 939
ags_audio_unit_devout_switch_buffer_flag, 940
ags_audio_unit_devout_test_flags, 939
ags_audio_unit_devout_unset_flags, 940
AGS_AUDIO_UNIT_PORT, 924
AGS_AUDIO_UNIT_PORT_CLASS, 924
AGS_AUDIO_UNIT_PORT_DEFAULT_CACHE_BUFFER_SIZE, 925

ags_audio_unit_port_find, 923
ags_audio_unit_port_flags_get_type, 925
AGS_AUDIO_UNIT_PORT_GET_CLASS, 924
AGS_AUDIO_UNIT_PORT_GET_OBJ_MUTEX, 922
ags_audio_unit_port_get_type, 925
ags_audio_unit_port_new, 924
ags_audio_unit_port_register, 923
ags_audio_unit_port_set_buffer_size, 924
ags_audio_unit_port_set_flags, 922
ags_audio_unit_port_set_format, 923
ags_audio_unit_port_set_pcm_channels, 924
ags_audio_unit_port_set_samplerate, 924
ags_audio_unit_port_test_flags, 922
ags_audio_unit_port_unregister, 923
ags_audio_unit_port_unset_flags, 922
AGS_AUDIO_UNIT_SERVER, 910
ags_audio_unit_server_add_client, 909
AGS_AUDIO_UNIT_SERVER_CLASS, 910
ags_audio_unit_server_connect_client, 910
ags_audio_unit_server_find_client, 908
ags_audio_unit_server_find_port, 909
ags_audio_unit_server_find_url, 908
AGS_AUDIO_UNIT_SERVER_GET_CLASS, 910
AGS_AUDIO_UNIT_SERVER_GET_OBJ_MUTEX, 907
ags_audio_unit_server_get_type, 911
ags_audio_unit_server_new, 910
ags_audio_unit_server_remove_client, 909
ags_audio_unit_server_set_flags, 907
ags_audio_unit_server_start_poll, 910
ags_audio_unit_server_test_flags, 907
ags_audio_unit_server_unset_flags, 907
ags_audio_unset_ability_flags, 967
ags_audio_unset_behaviour_flags, 968
ags_audio_unset_flags, 966
ags_audio_unset_staging_completed, 970
ags_audio_unset_staging_flags, 969
AgsAudioApplicationContext, 958
AgsAudioApplicationContextClass, 958
AgsAudioBufferUtil, 322
AgsAudioBufferUtilCopyMode, 316
AgsAudioBufferUtilFormat, 314
AgsAudioBufferUtilSamplerate, 315
AgsAudioClass, 1022
AgsAudioContainer, 1790
AgsAudioContainer:audio-channel, 1790
AgsAudioContainer:audio-signal, 1790
AgsAudioContainer:buffer-size, 1790
AgsAudioContainer:file-audio-channels, 1791
AgsAudioContainer:file-frame-count, 1791
AgsAudioContainer:file-samplerate, 1791
AgsAudioContainer:filename, 1791
AgsAudioContainer:format, 1791
AgsAudioContainer:instrument, 1792
AgsAudioContainer:preset, 1792
AgsAudioContainer:sample, 1792
AgsAudioContainer:samplerate, 1792
AgsAudioContainer:soundcard, 1792
AgsAudioContainer:wave, 1793
AgsAudioContainerClass, 1790
AgsAudioContainerManager, 1781
AgsAudioContainerManager:audio-container, 1781
AgsAudioContainerManagerClass, 1781
AgsAudioFile, 1812
AgsAudioFile:audio-channel, 1813
AgsAudioFile:audio-signal, 1813
AgsAudioFile:buffer-size, 1813
AgsAudioFile:file-audio-channels, 1813
AgsAudioFile:file-frame-count, 1814
AgsAudioFile:file-samplerate, 1814
AgsAudioFile:filename, 1814
AgsAudioFile:format, 1814
AgsAudioFile:samplerate, 1814
AgsAudioFile:soundcard, 1815
AgsAudioFile:wave, 1815
AgsAudioFileClass, 1812
AgsAudioFileLink, 1817
AgsAudioFileLink:audio-channel, 1817
AgsAudioFileLink:instrument, 1818
AgsAudioFileLink:preset, 1818
AgsAudioFileLink:sample, 1818
AgsAudioFileLink:timestamp, 1818
AgsAudioFileLinkClass, 1817
AgsAudioFileManager, 1796
AgsAudioFileManager:audio-file, 1797
AgsAudioFileManagerClass, 1796
AgsAudioFlags, 1020
AgsAudioLoop, 1455
AgsAudioLoop:play-audio, 1456
AgsAudioLoop:play-channel, 1456
AgsAudioLoopClass, 1456
AgsAudioLoopFlags, 1455
AgsAudioSignal, 1143
AgsAudioSignal::add-note, 1149
AgsAudioSignal::refresh-data, 1150
AgsAudioSignal::remove-note, 1150
AgsAudioSignal:attack, 1143
AgsAudioSignal:buffer-size, 1144
AgsAudioSignal:damping, 1144
AgsAudioSignal:delay, 1144
AgsAudioSignal:first-frame, 1144
AgsAudioSignal:format, 1145
AgsAudioSignal:frame-count, 1145
AgsAudioSignal:input-soundcard, 1145
AgsAudioSignal:input-soundcard-channel, 1145
AgsAudioSignal:last-frame, 1145
AgsAudioSignal:length, 1146
AgsAudioSignal:loop-end, 1146
AgsAudioSignal:loop-start, 1146
AgsAudioSignal:note, 1146
AgsAudioSignal:output-soundcard, 1146
AgsAudioSignal:output-soundcard-channel, 1147
AgsAudioSignal:recall-id, 1147
AgsAudioSignal:recycling, 1147
AgsAudioSignal:rt-template, 1147

AgsAudioSignal:samplerate, 1147
AgsAudioSignal:stream, 1148
AgsAudioSignal:stream-current, 1148
AgsAudioSignal:stream-end, 1148
AgsAudioSignal:template, 1148
AgsAudioSignal:timbre-end, 1148
AgsAudioSignal:timbre-start, 1149
AgsAudioSignal:vibration, 1149
AgsAudioSignal:word-size, 1149
AgsAudioSignalClass, 1143
AgsAudioSignalFlags, 1142
AgsAudioThread, 1464
AgsAudioThread:audio, 1465
AgsAudioThread:default-output-soundcard, 1465
AgsAudioThread:processing, 1465
AgsAudioThread:task-launcher, 1465
AgsAudioThreadClass, 1464
AgsAudioThreadFlags, 1463
AgsAudioThreadScopeData, 1464
AgsAudioUnitClient, 919
AgsAudioUnitClient:audio-unit-server, 919
AgsAudioUnitClient:client-name, 920
AgsAudioUnitClient:device, 920
AgsAudioUnitClient:port, 920
AgsAudioUnitClientClass, 919
AgsAudioUnitClientFlags, 919
AgsAudioUnitDevin, 934
AgsAudioUnitDevin:attack, 934
AgsAudioUnitDevin:audio-unit-client, 935
AgsAudioUnitDevin:audio-unit-port, 935
AgsAudioUnitDevin:bpm, 935
AgsAudioUnitDevin:buffer, 935
AgsAudioUnitDevin:buffer-size, 935
AgsAudioUnitDevin:delay-factor, 936
AgsAudioUnitDevin:device, 936
AgsAudioUnitDevin:dsp-channels, 936
AgsAudioUnitDevin:format, 936
AgsAudioUnitDevin:pcm-channels, 937
AgsAudioUnitDevin:samplerate, 937
AgsAudioUnitDevinAppBufferMode, 932
AgsAudioUnitDevinClass, 934
AgsAudioUnitDevinError, 934
AgsAudioUnitDevinFlags, 932
AgsAudioUnitDevinSyncFlags, 933
AgsAudioUnitDevout, 944
AgsAudioUnitDevout:attack, 944
AgsAudioUnitDevout:audio-unit-client, 945
AgsAudioUnitDevout:audio-unit-port, 945
AgsAudioUnitDevout:bpm, 945
AgsAudioUnitDevout:buffer, 945
AgsAudioUnitDevout:buffer-size, 945
AgsAudioUnitDevout:delay-factor, 946
AgsAudioUnitDevout:device, 946
AgsAudioUnitDevout:dsp-channels, 946
AgsAudioUnitDevout:format, 946
AgsAudioUnitDevout:pcm-channels, 947
AgsAudioUnitDevout:samplerate, 947
AgsAudioUnitDevoutAppBufferMode, 942
AgsAudioUnitDevoutClass, 944
AgsAudioUnitDevoutError, 944
AgsAudioUnitDevoutFlags, 942
AgsAudioUnitDevoutSyncFlags, 943
AgsAudioUnitPort, 926
AgsAudioUnitPort:audio-unit-client, 926
AgsAudioUnitPort:audio-unit-device, 926
AgsAudioUnitPort:port-name, 927
AgsAudioUnitPortClass, 926
AgsAudioUnitPortFlags, 925
AgsAudioUnitServer, 911
AgsAudioUnitServer:audio-unit-client, 911
AgsAudioUnitServer:default-audio-unit-client, 912
AgsAudioUnitServer:default-soundcard, 912
AgsAudioUnitServer:input-audio-unit-client, 912
AgsAudioUnitServer:url, 912
AgsAudioUnitServerClass, 911
AGS_AUTOMATION, 1231
AgsAutomation, 1234
AgsAutomation:acceleration, 1234
AgsAutomation:audio, 1234
AgsAutomation:channel-type, 1234
AgsAutomation:control-name, 1235
AgsAutomation:default-value, 1235
AgsAutomation:line, 1235
AgsAutomation:lower, 1235
AgsAutomation:port, 1236
AgsAutomation:steps, 1236
AgsAutomation:timestamp, 1236
AgsAutomation:upper, 1236
ags_automation_add, 1213
ags_automation_add_acceleration, 1221
ags_automation_add_all_to_selection, 1226
ags_automation_add_point_to_selection, 1224
ags_automation_add_region_to_selection, 1225
AGS_AUTOMATION_CLASS, 1231
AGS_AUTOMATION_CLIPBOARD_FORMAT, 1233
AGS_AUTOMATION_CLIPBOARD_LEGACY_FORMAT, 1233
AGS_AUTOMATION_CLIPBOARD_TYPE, 1233
AGS_AUTOMATION_CLIPBOARD_VERSION, 1233
ags_automation_copy_selection, 1226
ags_automation_cut_selection, 1226
AGS_AUTOMATION_DEFAULT_BPM, 1232
AGS_AUTOMATION_DEFAULT_DURATION, 1232
AGS_AUTOMATION_DEFAULT_JIFFIE, 1232
AGS_AUTOMATION_DEFAULT_LENGTH, 1232
AGS_AUTOMATION_DEFAULT_OFFSET, 1233
AGS_AUTOMATION_DEFAULT_PRECISION, 1233
ags_automation_find_channel_type_with_control_name, 1229
ags_automation_find_near_timestamp, 1212
ags_automation_find_near_timestamp_extended, 1213
ags_automation_find_point, 1223
ags_automation_find_port, 1212
ags_automation_find_region, 1223
ags_automation_find_specifier, 1229

ags_automation_find_specifier_with_type_and_line, 1230
ags_automation_flags_get_type, 1232
ags_automation_free_all_selection, 1224
ags_automation_free_selection, 1224
ags_automation_get_acceleration, 1220
ags_automation_get_audio, 1214
ags_automation_get_channel_type, 1214
AGS_AUTOMATION_GET_CLASS, 1231
ags_automation_get_control_name, 1216
ags_automation_get_default_value, 1219
ags_automation_get_line, 1215
ags_automation_get_lower, 1218
AGS_AUTOMATION_GET_OBJ_MUTEX, 1210
ags_automation_get_obj_mutex, 1210
ags_automation_get_port, 1220
ags_automation_get_selection, 1222
ags_automation_get_specifier_unique, 1228
ags_automation_get_specifier_unique_with_channel_type, 1228
ags_automation_get_steps, 1217
ags_automation_get_timestamp, 1216
ags_automation_get_type, 1232
ags_automation_get_upper, 1218
ags_automation_get_value, 1230
ags_automation_insert_from_clipboard, 1227
ags_automation_insert_from_clipboard_extended, 1227
ags_automation_is_acceleration_selected, 1222
AGS_AUTOMATION_MAXIMUM_ACCELERATION_LENGTH, 1232
AGS_AUTOMATION_MAXIMUM_STEPS, 1233
AGS_AUTOMATION_MINIMUM_ACCELERATION_LENGTH, 1232
ags_automation_new, 1231
ags_automation_remove_acceleration, 1221
ags_automation_remove_acceleration_at_position, 1222
ags_automation_remove_point_from_selection, 1225
ags_automation_remove_region_from_selection, 1225
ags_automation_set_acceleration, 1221
ags_automation_set_audio, 1214
ags_automation_set_channel_type, 1215
ags_automation_set_control_name, 1217
ags_automation_set_default_value, 1219
ags_automation_set_flags, 1211
ags_automation_set_line, 1215
ags_automation_set_lower, 1219
ags_automation_set_port, 1220
ags_automation_set_steps, 1217
ags_automation_set_timestamp, 1216
ags_automation_set_upper, 1218
ags_automation_sort_func, 1213
ags_automation_test_flags, 1211
AGS_AUTOMATION_TICS_PER_BEAT, 1232
ags_automation_unset_flags, 1211
AgsAutomationClass, 1234
AgsAutomationFlags, 1233

B

AGS_BASE_PLUGIN, 17
ags_base_plugin_activate, 16
ags_base_plugin_apply_port_group_by_prefix, 14
AGS_BASE_PLUGIN_CLASS, 17
ags_base_plugin_connect_port, 15
ags_base_plugin_deactivate, 16
ags_base_plugin_find_effect, 13
ags_base_plugin_find_filename, 12
ags_base_plugin_find_ui_effect_index, 13
ags_base_plugin_flags_get_type, 18
AGS_BASE_PLUGIN_GET_CLASS, 17
ags_base_plugin_get_effect, 7
ags_base_plugin_get_effect_index, 8
ags_base_plugin_get_filename, 6
AGS_BASE_PLUGIN_GET_OBJ_MUTEX, 5
ags_base_plugin_get_obj_mutex, 5
ags_base_plugin_get_plugin_port, 9
ags_base_plugin_get_plugin_so, 8
ags_base_plugin_get_type, 18
ags_base_plugin_get_ui_effect, 10
ags_base_plugin_get_ui_effect_index, 11
ags_base_plugin_get_ui_filename, 9
ags_base_plugin_get_ui_plugin, 12
ags_base_plugin_get_ui_plugin_so, 11
ags_base_plugin_instantiate, 14
ags_base_plugin_instantiate_with_params, 15
ags_base_plugin_load_plugin, 17
ags_base_plugin_new, 17
ags_base_plugin_run, 16
ags_base_plugin_set_effect, 7
ags_base_plugin_set_effect_index, 8
ags_base_plugin_set_filename, 7
ags_base_plugin_set_flags, 6
ags_base_plugin_set_plugin_port, 9
ags_base_plugin_set_plugin_so, 9
ags_base_plugin_set_ui_effect, 10
ags_base_plugin_set_ui_effect_index, 11
ags_base_plugin_set_ui_filename, 10
ags_base_plugin_set_ui_plugin, 12
ags_base_plugin_set_ui_plugin_so, 12
ags_base_plugin_sort, 14
ags_base_plugin_sort_func, 13
ags_base_plugin_test_flags, 5
ags_base_plugin_unset_flags, 6
AgsBasePlugin, 18
AgsBasePlugin::activate, 21
AgsBasePlugin::connect-port, 22
AgsBasePlugin::deactivate, 22
AgsBasePlugin::instantiate, 23
AgsBasePlugin::instantiate-with-params, 23
AgsBasePlugin::load-plugin, 24
AgsBasePlugin::run, 24
AgsBasePlugin:effect, 19
AgsBasePlugin:effect-index, 19
AgsBasePlugin:filename, 20
AgsBasePlugin:plugin-port, 20
AgsBasePlugin:plugin-so, 20
AgsBasePlugin:ui-effect, 20

AgsBasePlugin:ui-effect-index, 20
AgsBasePlugin:ui-filename, 21
AgsBasePlugin:ui-plugin, 21
AgsBasePlugin:ui-plugin-so, 21
AgsBasePlugin:uuid, 21
AgsBasePluginClass, 19
AgsBasePluginFlags, 18
AGS_BUFFER, 1271
AgsBuffer, 1272
AgsBuffer:buffer-size, 1272
AgsBuffer:data, 1272
AgsBuffer:format, 1272
AgsBuffer:samplerate, 1273
AgsBuffer:selection-x0, 1273
AgsBuffer:selection-x1, 1273
AgsBuffer:x, 1273
AGS_BUFFER_CLASS, 1271
ags_buffer_duplicate, 1270
ags_buffer_flags_get_type, 1271
ags_buffer_get_buffer_size, 1268
AGS_BUFFER_GET_CLASS, 1271
ags_buffer_get_data, 1270
ags_buffer_get_format, 1269
AGS_BUFFER_GET_OBJ_MUTEX, 1265
ags_buffer_get_obj_mutex, 1265
ags_buffer_get_samplerate, 1268
ags_buffer_get_type, 1271
ags_buffer_get_x, 1267
ags_buffer_lock, 1265
ags_buffer_new, 1270
ags_buffer_set_buffer_size, 1269
ags_buffer_set_flags, 1266
ags_buffer_set_format, 1269
ags_buffer_set_samplerate, 1268
ags_buffer_set_x, 1268
ags_buffer_sort_func, 1267
ags_buffer_test_flags, 1266
ags_buffer_unlock, 1266
ags_buffer_unset_flags, 1267
AgsBufferClass, 1272
AgsBufferFlags, 1271

C

AGS_CANCEL_AUDIO, 2315
AGS_CANCEL_AUDIO_CLASS, 2315
AGS_CANCEL_AUDIO_GET_CLASS, 2315
ags_cancel_audio_get_type, 2315
ags_cancel_audio_new, 2314
AGS_CANCEL_CHANNEL, 2318
AGS_CANCEL_CHANNEL_CLASS, 2318
AGS_CANCEL_CHANNEL_GET_CLASS, 2318
ags_cancel_channel_get_type, 2318
ags_cancel_channel_new, 2317
AgsCancelAudio, 2316
AgsCancelAudio:audio, 2316
AgsCancelAudio:sound-scope, 2316
AgsCancelAudioClass, 2316
AgsCancelChannel, 2319
AgsCancelChannel:channel, 2319
AgsCancelChannel:sound-scope, 2319
AgsCancelChannelClass, 2319
AGS_CHANNEL, 1080
AgsChannel, 1083
AgsChannel::cancel-recall, 1091
AgsChannel::check-scope, 1091
AgsChannel::cleanup-recall, 1092
AgsChannel::done-recall, 1092
AgsChannel::duplicate-recall, 1093
AgsChannel::init-recall, 1093
AgsChannel::play-recall, 1093
AgsChannel::recursive-run-stage, 1094
AgsChannel::recycling-changed, 1094
AgsChannel::resolve-recall, 1095
AgsChannel::start, 1095
AgsChannel::stop, 1096
AgsChannel:absolute-key, 1084
AgsChannel:audio, 1084
AgsChannel:audio-channel, 1084
AgsChannel:buffer-size, 1085
AgsChannel:first-recycling, 1085
AgsChannel:format, 1085
AgsChannel:input-soundcard, 1085
AgsChannel:input-soundcard-channel, 1086
AgsChannel:key, 1086
AgsChannel:last-recycling, 1086
AgsChannel:line, 1086
AgsChannel:link, 1087
AgsChannel:midi-note, 1087
AgsChannel:next, 1087
AgsChannel:next-pad, 1087
AgsChannel:note-frequency, 1087
AgsChannel:note-key, 1088
AgsChannel:octave, 1088
AgsChannel:output-soundcard, 1088
AgsChannel:output-soundcard-channel, 1088
AgsChannel:pad, 1089
AgsChannel:pattern, 1089
AgsChannel:play, 1089
AgsChannel:playback, 1089
AgsChannel:prev, 1089
AgsChannel:prev-pad, 1090
AgsChannel:recall, 1090
AgsChannel:recall-container, 1090
AgsChannel:recall-id, 1090
AgsChannel:recycling-context, 1090
AgsChannel:samplerate, 1091
ags_channel_add_pattern, 1068
ags_channel_add_recall, 1074
ags_channel_add_recall_container, 1072
ags_channel_add_recall_id, 1070
ags_channel_cancel_recall, 1076
ags_channel_check_scope, 1078
AGS_CHANNEL_CLASS, 1081
ags_channel_cleanup_recall, 1077

ags_channel_collect_all_channel_ports, 1078
ags_channel_collect_all_channel_ports_by_specifier_and_context, 1079
AGS_CHANNEL_DEFAULT_MIDI_NOTE, 1083
AGS_CHANNEL_DEFAULT_NOTE_FREQUENCY, 1082
AGS_CHANNEL_DEFAULT_OCTAVE, 1081
AGS_CHANNEL_DEFAULT_OCTAVE_SEMITONE, 1082
AGS_CHANNEL_DEFAULT_SEMITONE, 1082
ags_channel_done_recall, 1076
ags_channel_duplicate_recall, 1075
AGS_CHANNEL_ERROR, 1083
ags_channel_error_quark, 1046
ags_channel_first, 1054
ags_channel_first_with_recycling, 1056
ags_channel_flags_get_type, 1081
ags_channel_get_absolute_key, 1067
ags_channel_get_audio, 1052
ags_channel_get_audio_channel, 1065
ags_channel_get_buffer_size, 1063
AGS_CHANNEL_GET_CLASS, 1081
ags_channel_get_format, 1063
ags_channel_get_input_soundcard, 1061
ags_channel_get_input_soundcard_channel, 1061
ags_channel_get_key, 1066
ags_channel_get_level, 1079
ags_channel_get_line, 1065
ags_channel_get_link, 1058
AGS_CHANNEL_GET_OBJ_MUTEX, 1046
ags_channel_get_obj_mutex, 1046
ags_channel_get_octave, 1066
ags_channel_get_output_soundcard, 1059
ags_channel_get_output_soundcard_channel, 1060
ags_channel_get_pad, 1064
ags_channel_get_pattern, 1068
ags_channel_get_play, 1072
AGS_CHANNEL_GET_PLAY_MUTEX, 1046
ags_channel_get_play_mutex, 1047
ags_channel_get_playback, 1069
ags_channel_get_recall, 1073
ags_channel_get_recall_container, 1071
ags_channel_get_recall_id, 1070
AGS_CHANNEL_GET_RECALL_MUTEX, 1046
ags_channel_get_recall_mutex, 1047
ags_channel_get_samplerate, 1062
ags_channel_get_type, 1081
ags_channel_init_recall, 1075
ags_channel_insert_recall, 1074
ags_channel_last, 1055
ags_channel_last_with_recycling, 1057
AGS_CHANNEL_MAXIMUM_MIDI_NOTE, 1082
AGS_CHANNEL_MAXIMUM_NOTE_FREQUENCY, 1082
AGS_CHANNEL_MAXIMUM_OCTAVE, 1081
AGS_CHANNEL_MAXIMUM_OCTAVE_SEMITONE, 1082
AGS_CHANNEL_MAXIMUM_SEMITONE, 1082
AGS_CHANNEL_MINIMUM_MIDI_NOTE, 1082
AGS_CHANNEL_MINIMUM_NOTE_FREQUENCY, 1082
AGS_CHANNEL_MINIMUM_OCTAVE, 1081
AGS_CHANNEL_MINIMUM_OCTAVE_SEMITONE, 1082
AGS_CHANNEL_MINIMUM_SEMITONE, 1082
ags_channel_new, 1080
ags_channel_next, 1053
ags_channel_next_pad, 1054
ags_channel_next_with_recycling, 1057
ags_channel_nth, 1055
ags_channel_pad_first, 1055
ags_channel_pad_last, 1056
ags_channel_pad_nth, 1056
ags_channel_play_recall, 1076
ags_channel_prev, 1053
ags_channel_prev_pad, 1054
ags_channel_prev_with_recycling, 1057
ags_channel_recursive_run_stage, 1080
ags_channel_recursive_set_property, 1079
ags_channel_recycling_changed, 1059
ags_channel_remove_pattern, 1069
ags_channel_remove_recall, 1074
ags_channel_remove_recall_container, 1072
ags_channel_remove_recall_id, 1071
ags_channel_reset_recycling, 1059
ags_channel_resolve_recall, 1075
ags_channel_set_ability_flags, 1049
ags_channel_set_absolute_key, 1067
ags_channel_set_audio, 1053
ags_channel_set_audio_channel, 1065
ags_channel_set_behaviour_flags, 1050
ags_channel_set_buffer_size, 1063
ags_channel_set_flags, 1048
ags_channel_set_format, 1064
ags_channel_set_input_soundcard, 1061
ags_channel_set_input_soundcard_channel, 1062
ags_channel_set_key, 1067
ags_channel_set_line, 1066
ags_channel_set_link, 1058
ags_channel_set_octave, 1066
ags_channel_set_output_soundcard, 1060
ags_channel_set_output_soundcard_channel, 1060
ags_channel_set_pad, 1064
ags_channel_set_pattern, 1068
ags_channel_set_play, 1073
ags_channel_set_playback, 1069
ags_channel_set_recall, 1073
ags_channel_set_recall_container, 1071
ags_channel_set_recall_id, 1070
ags_channel_set_samplerate, 1062
ags_channel_set_staging_completed, 1052
ags_channel_set_staging_flags, 1051
ags_channel_start, 1077
ags_channel_stop, 1077
ags_channel_test_ability_flags, 1048
ags_channel_test_behaviour_flags, 1049
ags_channel_test_flags, 1047
ags_channel_test_staging_completed, 1051
ags_channel_test_staging_flags, 1050
AGS_CHANNEL_THREAD, 1471

AGS_CHANNEL_THREAD_CLASS, 1471
AGS_CHANNEL_THREAD_DEFAULT_JIFFIE, 1471
AGS_CHANNEL_THREAD_GET_CLASS, 1471
ags_channel_thread_get_do_fx_staging, 1469
ags_channel_thread_get_processing, 1468
ags_channel_thread_get_staging_program, 1470
ags_channel_thread_get_task_launcher, 1468
ags_channel_thread_get_type, 1471
ags_channel_thread_new, 1470
ags_channel_thread_set_do_fx_staging, 1469
ags_channel_thread_set_processing, 1468
ags_channel_thread_set_sound_scope, 1469
ags_channel_thread_set_staging_program, 1470
ags_channel_thread_set_status_flags, 1467
ags_channel_thread_test_status_flags, 1467
ags_channel_thread_unset_status_flags, 1467
ags_channel_unset_ability_flags, 1049
ags_channel_unset_behaviour_flags, 1050
ags_channel_unset_flags, 1048
ags_channel_unset_staging_completed, 1052
ags_channel_unset_staging_flags, 1051
AgsChannelClass, 1083
AgsChannelError, 1083
AgsChannelFlags, 1083
AgsChannelThread, 1473
AgsChannelThread:channel, 1473
AgsChannelThread:default-output-soundcard, 1473
AgsChannelThread:processing, 1473
AgsChannelThread:task-launcher, 1474
AgsChannelThreadClass, 1473
AgsChannelThreadFlags, 1472
ags_char_buffer_util_copy_buffer_to_buffer, 170
ags_char_buffer_util_copy_cbuffer_to_double, 169
ags_char_buffer_util_copy_cbuffer_to_float, 169
ags_char_buffer_util_copy_cbuffer_to_s16, 167
ags_char_buffer_util_copy_cbuffer_to_s24, 167
ags_char_buffer_util_copy_cbuffer_to_s32, 168
ags_char_buffer_util_copy_cbuffer_to_s64, 168
ags_char_buffer_util_copy_cbuffer_to_s8, 166
ags_char_buffer_util_copy_double_to_cbuffer, 166
ags_char_buffer_util_copy_float_to_cbuffer, 165
ags_char_buffer_util_copy_s16_to_cbuffer, 163
ags_char_buffer_util_copy_s24_to_cbuffer, 164
ags_char_buffer_util_copy_s32_to_cbuffer, 164
ags_char_buffer_util_copy_s64_to_cbuffer, 165
ags_char_buffer_util_copy_s8_to_cbuffer, 162
ags_char_buffer_util_get_type, 170
AgsCharBufferUtil, 172
AgsCharBufferUtilCopyMode, 170
AGS_CHORUS_UTIL, 188
ags_chorus_util_alloc, 174
ags_chorus_util_compute, 188
ags_chorus_util_compute_complex, 188
ags_chorus_util_compute_double, 188
ags_chorus_util_compute_float, 187
ags_chorus_util_compute_s16, 186
ags_chorus_util_compute_s24, 187
ags_chorus_util_compute_s32, 187
ags_chorus_util_compute_s64, 187
ags_chorus_util_compute_s8, 186
ags_chorus_util_copy, 174
AGS_CHORUS_UTIL_DEFAULT_DELAY, 189
AGS_CHORUS_UTIL_DEFAULT_DEPTH, 189
AGS_CHORUS_UTIL_DEFAULT_HISTORY_BUFFER_LENGT
189
AGS_CHORUS_UTIL_DEFAULT_LFO_FREQUENCY, 189
AGS_CHORUS_UTIL_DEFAULT_LFO_OSCILLATOR, 189
AGS_CHORUS_UTIL_DEFAULT_MIX, 189
AGS_CHORUS_UTIL_DEFAULT_PITCH_MIX_BUFFER_SIZE
189
ags_chorus_util_free, 174
ags_chorus_util_get_base_key, 180
ags_chorus_util_get_buffer_length, 177
ags_chorus_util_get_delay, 184
ags_chorus_util_get_depth, 183
ags_chorus_util_get_destination, 176
ags_chorus_util_get_destination_stride, 177
ags_chorus_util_get_format, 178
ags_chorus_util_get_input_volume, 180
ags_chorus_util_get_lfo_frequency, 182
ags_chorus_util_get_lfo_oscillator, 182
ags_chorus_util_get_mix, 183
ags_chorus_util_get_offset, 179
ags_chorus_util_get_output_volume, 181
ags_chorus_util_get_pitch_type, 185
ags_chorus_util_get_pitch_util, 185
ags_chorus_util_get_samplerate, 178
ags_chorus_util_get_source, 175
ags_chorus_util_get_source_stride, 175
ags_chorus_util_get_type, 189
ags_chorus_util_set_base_key, 180
ags_chorus_util_set_buffer_length, 178
ags_chorus_util_set_delay, 184
ags_chorus_util_set_depth, 183
ags_chorus_util_set_destination, 176
ags_chorus_util_set_destination_stride, 177
ags_chorus_util_set_format, 178
ags_chorus_util_set_input_volume, 181
ags_chorus_util_set_lfo_frequency, 183
ags_chorus_util_set_lfo_oscillator, 182
ags_chorus_util_set_mix, 184
ags_chorus_util_set_offset, 179
ags_chorus_util_set_output_volume, 181
ags_chorus_util_set_pitch_type, 185
ags_chorus_util_set_pitch_util, 186
ags_chorus_util_set_samplerate, 179
ags_chorus_util_set_source, 175
ags_chorus_util_set_source_stride, 176
AgsChorusUtil, 190
AGS_CLEAR_AUDIO_SIGNAL, 2321
AGS_CLEAR_AUDIO_SIGNAL_CLASS, 2321
AGS_CLEAR_AUDIO_SIGNAL_GET_CLASS, 2321
ags_clear_audio_signal_get_type, 2321
ags_clear_audio_signal_new, 2320

AGS_CLEAR_BUFFER, 2323
AGS_CLEAR_BUFFER_CLASS, 2323
AGS_CLEAR_BUFFER_GET_CLASS, 2323
ags_clear_buffer_get_type, 2324
ags_clear_buffer_new, 2323
AgsClearAudioSignal, 2321
AgsClearAudioSignal:audio-signal, 2322
AgsClearAudioSignalClass, 2322
AgsClearBuffer, 2324
AgsClearBuffer:device, 2324
AgsClearBufferClass, 2324
AGS_CORE_AUDIO_CLIENT, 868
ags_core_audio_client_activate, 867
ags_core_audio_client_add_device, 866
ags_core_audio_client_add_port, 867
AGS_CORE_AUDIO_CLIENT_CLASS, 868
ags_core_audio_client_deactivate, 868
ags_core_audio_client_find, 866
ags_core_audio_client_find_uuid, 865
ags_core_audio_client_flags_get_type, 869
AGS_CORE_AUDIO_CLIENT_GET_CLASS, 869
AGS_CORE_AUDIO_CLIENT_GET_OBJ_MUTEX, 864
ags_core_audio_client_get_type, 869
ags_core_audio_client_new, 868
ags_core_audio_client_open, 866
ags_core_audio_client_remove_device, 867
ags_core_audio_client_remove_port, 867
ags_core_audio_client_set_flags, 865
ags_core_audio_client_test_flags, 864
ags_core_audio_client_unset_flags, 865
AGS_CORE_AUDIO_DEVIN, 882
ags_core_audio_devin_adjust_delay_and_attack, 881
AGS_CORE_AUDIO_DEVIN_CLASS, 882
AGS_CORE_AUDIO_DEVIN_DEFAULT_APP_BUFFER_SIZE, 883
AGS_CORE_AUDIO_DEVIN_ERROR, 884
ags_core_audio_devin_error_quark, 880
ags_core_audio_devin_flags_get_type, 882
AGS_CORE_AUDIO_DEVIN_GET_CLASS, 882
AGS_CORE_AUDIO_DEVIN_GET_OBJ_MUTEX, 879
ags_core_audio_devin_get_type, 882
ags_core_audio_devin_new, 882
ags_core_audio_devin_realloc_buffer, 881
ags_core_audio_devin_set_flags, 880
ags_core_audio_devin_switch_buffer_flag, 881
ags_core_audio_devin_test_flags, 880
ags_core_audio_devin_unset_flags, 880
AGS_CORE_AUDIO_DEVOUT, 891
ags_core_audio_devout_adjust_delay_and_attack, 891
AGS_CORE_AUDIO_DEVOUT_CLASS, 891
AGS_CORE_AUDIO_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 892
AGS_CORE_AUDIO_DEVOUT_ERROR, 894
ags_core_audio_devout_error_quark, 889
ags_core_audio_devout_flags_get_type, 892
AGS_CORE_AUDIO_DEVOUT_GET_CLASS, 892
AGS_CORE_AUDIO_DEVOUT_GET_OBJ_MUTEX, 889
ags_core_audio_devout_get_type, 892
ags_core_audio_devout_new, 891
ags_core_audio_devout_realloc_buffer, 891
ags_core_audio_devout_set_flags, 890
ags_core_audio_devout_switch_buffer_flag, 890
ags_core_audio_devout_test_flags, 889
ags_core_audio_devout_unset_flags, 890
AGS_CORE_AUDIO_MIDIIN, 900
AGS_CORE_AUDIO_MIDIIN_CLASS, 900
AGS_CORE_AUDIO_MIDIIN_DEFAULT_BUFFER_SIZE, 901
AGS_CORE_AUDIO_MIDIIN_ERROR, 903
ags_core_audio_midiin_error_quark, 899
ags_core_audio_midiin_flags_get_type, 901
AGS_CORE_AUDIO_MIDIIN_GET_CLASS, 901
AGS_CORE_AUDIO_MIDIIN_GET_OBJ_MUTEX, 899
ags_core_audio_midiin_get_type, 901
ags_core_audio_midiin_new, 900
ags_core_audio_midiin_set_flags, 899
ags_core_audio_midiin_switch_buffer_flag, 900
ags_core_audio_midiin_test_flags, 899
ags_core_audio_midiin_unset_flags, 900
AGS_CORE_AUDIO_PORT, 875
AGS_CORE_AUDIO_PORT_CLASS, 875
AGS_CORE_AUDIO_PORT_DEFAULT_CACHE_BUFFER_SIZ 876
ags_core_audio_port_find, 873
ags_core_audio_port_flags_get_type, 876
AGS_CORE_AUDIO_PORT_GET_CLASS, 876
ags_core_audio_port_get_latency, 875
AGS_CORE_AUDIO_PORT_GET_OBJ_MUTEX, 872
ags_core_audio_port_get_type, 876
ags_core_audio_port_new, 875
ags_core_audio_port_register, 874
ags_core_audio_port_set_buffer_size, 874
ags_core_audio_port_set_cache_buffer_size, 875
ags_core_audio_port_set_flags, 873
ags_core_audio_port_set_format, 874
ags_core_audio_port_set_pcm_channels, 874
ags_core_audio_port_set_samplerate, 874
ags_core_audio_port_test_flags, 872
ags_core_audio_port_unregister, 874
ags_core_audio_port_unset_flags, 873
AGS_CORE_AUDIO_SERVER, 861
ags_core_audio_server_add_client, 860
AGS_CORE_AUDIO_SERVER_CLASS, 861
ags_core_audio_server_connect_client, 860
ags_core_audio_server_find_client, 859
ags_core_audio_server_find_port, 859
ags_core_audio_server_find_url, 858
AGS_CORE_AUDIO_SERVER_GET_CLASS, 861
AGS_CORE_AUDIO_SERVER_GET_OBJ_MUTEX, 857
ags_core_audio_server_get_type, 861
ags_core_audio_server_new, 861
ags_core_audio_server_remove_client, 860
ags_core_audio_server_set_flags, 858
ags_core_audio_server_start_poll, 860

ags_core_audio_server_test_flags, 857
ags_core_audio_server_unset_flags, 858
AgsCore_AudioDevinAppBufferMode, 883
AgsCore_AudioDevoutAppBufferMode, 893
AgsCoreAudioClient, 870
AgsCoreAudioClient:client-name, 870
AgsCoreAudioClient:core-audio-server, 870
AgsCoreAudioClient:device, 870
AgsCoreAudioClient:port, 871
AgsCoreAudioClientClass, 870
AgsCoreAudioClientFlags, 869
AgsCoreAudioDevin, 885
AgsCoreAudioDevin:attack, 885
AgsCoreAudioDevin:bpm, 885
AgsCoreAudioDevin:buffer, 885
AgsCoreAudioDevin:buffer-size, 886
AgsCoreAudioDevin:core-audio-client, 886
AgsCoreAudioDevin:core-audio-port, 886
AgsCoreAudioDevin:delay-factor, 886
AgsCoreAudioDevin:device, 886
AgsCoreAudioDevin:dsp-channels, 887
AgsCoreAudioDevin:format, 887
AgsCoreAudioDevin:pcm-channels, 887
AgsCoreAudioDevin:samplerate, 887
AgsCoreAudioDevinClass, 885
AgsCoreAudioDevinError, 884
AgsCoreAudioDevinFlags, 883
AgsCoreAudioDevinSyncFlags, 884
AgsCoreAudioDevout, 894
AgsCoreAudioDevout:attack, 895
AgsCoreAudioDevout:bpm, 895
AgsCoreAudioDevout:buffer, 895
AgsCoreAudioDevout:buffer-size, 895
AgsCoreAudioDevout:core-audio-client, 896
AgsCoreAudioDevout:core-audio-port, 896
AgsCoreAudioDevout:delay-factor, 896
AgsCoreAudioDevout:device, 896
AgsCoreAudioDevout:dsp-channels, 896
AgsCoreAudioDevout:format, 897
AgsCoreAudioDevout:pcm-channels, 897
AgsCoreAudioDevout:samplerate, 897
AgsCoreAudioDevoutClass, 894
AgsCoreAudioDevoutError, 894
AgsCoreAudioDevoutFlags, 892
AgsCoreAudioDevoutSyncFlags, 893
AgsCoreAudioMidiin, 903
AgsCoreAudioMidiin:attack, 904
AgsCoreAudioMidiin:bpm, 904
AgsCoreAudioMidiin:buffer, 904
AgsCoreAudioMidiin:core-audio-client, 904
AgsCoreAudioMidiin:core-audio-port, 905
AgsCoreAudioMidiin:delay-factor, 905
AgsCoreAudioMidiin:device, 905
AgsCoreAudioMidiinAppBufferMode, 902
AgsCoreAudioMidiinClass, 904
AgsCoreAudioMidiinError, 903
AgsCoreAudioMidiinFlags, 901
AgsCoreAudioMidiinSyncFlags, 902
AgsCoreAudioPort, 877
AgsCoreAudioPort:core-audio-client, 877
AgsCoreAudioPort:core-audio-device, 878
AgsCoreAudioPort:port-name, 878
AgsCoreAudioPortClass, 877
AgsCoreAudioPortFlags, 876
AgsCoreAudioServer, 862
AgsCoreAudioServer:core-audio-client, 862
AgsCoreAudioServer:default-core-audio-client, 862
AgsCoreAudioServer:default-soundcard, 862
AgsCoreAudioServer:url, 863
AgsCoreAudioServerClass, 862
AGS_CROP_NOTE, 2327
AGS_CROP_NOTE_CLASS, 2327
AGS_CROP_NOTE_DEFAULT_X_LENGTH, 2327
AGS_CROP_NOTE_GET_CLASS, 2327
ags_crop_note_get_type, 2327
ags_crop_note_new, 2326
ags_crop_note_set_selection, 2326
AgsCropNote, 2327
AgsCropNote:absolute, 2328
AgsCropNote:audio, 2328
AgsCropNote:do-resize, 2328
AgsCropNote:in-place, 2328
AgsCropNote:notation, 2329
AgsCropNote:selection, 2329
AgsCropNote:x-crop, 2329
AgsCropNote:x-padding, 2329
AgsCropNoteClass, 2328

D

AgsDLS2Level, 1852
AGS_DSSI_MANAGER, 62
AGS_DSSI_MANAGER_CLASS, 62
ags_dssi_manager_find_dssi_plugin, 59
ags_dssi_manager_find_dssi_plugin_with_fallback, 59
AGS_DSSI_MANAGER_GET_CLASS, 62
ags_dssi_manager_get_default_path, 58
ags_dssi_manager_get_dssi_plugin, 61
ags_dssi_manager_get_filenames, 59
ags_dssi_manager_get_instance, 61
AGS_DSSI_MANAGER_GET_OBJ_MUTEX, 58
ags_dssi_manager_get_type, 62
ags_dssi_manager_load_blacklist, 60
ags_dssi_manager_load_default_directory, 60
ags_dssi_manager_load_file, 60
ags_dssi_manager_new, 61
ags_dssi_manager_set_default_path, 58
AGS_DSSI_PLUGIN, 55
ags_dssi_plugin_change_program, 54
AGS_DSSI_PLUGIN_CLASS, 55
AGS_DSSI_PLUGIN_DESCRIPTOR, 54
AGS_DSSI_PLUGIN_GET_CLASS, 55
ags_dssi_plugin_get_type, 55
ags_dssi_plugin_new, 54
AgsDssiManager, 62

AgsDssiManagerClass, 62
AgsDssiPlugin, 56
AgsDssiPlugin::change-program, 56
AgsDssiPlugin:program, 56
AgsDssiPlugin:unique-id, 56
AgsDssiPluginClass, 56

E
AGS_EFFECTS_DEFAULT_VERSION, 957
AGS_ENVELOPE_UTIL, 341
ags_envelope_util_alloc, 333
ags_envelope_util_compute, 340
ags_envelope_util_compute_complex, 340
ags_envelope_util_compute_double, 340
ags_envelope_util_compute_float, 340
ags_envelope_util_compute_s16, 339
ags_envelope_util_compute_s24, 339
ags_envelope_util_compute_s32, 339
ags_envelope_util_compute_s64, 339
ags_envelope_util_compute_s8, 338
ags_envelope_util_copy, 333
AGS_ENVELOPE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT, 341
AGS_ENVELOPE_UTIL_DEFAULT_FORMAT, 341
ags_envelope_util_free, 333
ags_envelope_util_get_amount, 338
ags_envelope_util_get_buffer_length, 336
ags_envelope_util_get_destination, 333
ags_envelope_util_get_destination_stride, 334
ags_envelope_util_get_format, 337
ags_envelope_util_get_source, 335
ags_envelope_util_get_source_stride, 335
ags_envelope_util_get_type, 341
ags_envelope_util_get_volume, 337
ags_envelope_util_set_amount, 338
ags_envelope_util_set_buffer_length, 336
ags_envelope_util_set_destination, 334
ags_envelope_util_set_destination_stride, 334
ags_envelope_util_set_format, 337
ags_envelope_util_set_source, 335
ags_envelope_util_set_source_stride, 336
ags_envelope_util_set_volume, 338
AgsEnvelopeUtil, 341
AGS_EXPORT_OUTPUT, 2331
AGS_EXPORT_OUTPUT_CLASS, 2331
AGS_EXPORT_OUTPUT_GET_CLASS, 2331
ags_export_output_get_type, 2332
ags_export_output_new, 2331
AGS_EXPORT_THREAD, 1476
AGS_EXPORT_THREAD_CLASS, 1476
AGS_EXPORT_THREAD_DEFAULT_JIFFIE, 1476
ags_export_thread_find_soundcard, 1475
AGS_EXPORT_THREAD_GET_CLASS, 1476
ags_export_thread_get_type, 1476
ags_export_thread_new, 1475
AgsExportOutput, 2332
AgsExportOutput:export-thread, 2333
AgsExportOutput:filename, 2333
AgsExportOutput:format, 2333
AgsExportOutput:live-performance, 2333
AgsExportOutput:soundcard, 2333
AgsExportOutput:tic, 2334
AgsExportOutputClass, 2332
AgsExportOutputFormat, 2332
AgsExportThread, 1477
AgsExportThread:audio-file, 1477
AgsExportThread:soundcard, 1477
AgsExportThread:tic, 1477
AgsExportThreadClass, 1477
AgsExportThreadFlags, 1476

F
AGS_FAST_PITCH_UTIL, 220
ags_fast_pitch_util_alloc, 212
ags_fast_pitch_util_copy, 212
AGS_FAST_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE, 221
ags_fast_pitch_util_free, 212
ags_fast_pitch_util_get_base_key, 217
ags_fast_pitch_util_get_buffer_length, 215
ags_fast_pitch_util_get_destination, 214
ags_fast_pitch_util_get_destination_stride, 214
ags_fast_pitch_util_get_format, 216
ags_fast_pitch_util_get_samplerate, 216
ags_fast_pitch_util_get_source, 213
ags_fast_pitch_util_get_source_stride, 213
ags_fast_pitch_util_get_tuning, 218
ags_fast_pitch_util_get_type, 221
ags_fast_pitch_util_pitch, 220
ags_fast_pitch_util_pitch_complex, 220
ags_fast_pitch_util_pitch_double, 220
ags_fast_pitch_util_pitch_float, 219
ags_fast_pitch_util_pitch_s16, 218
ags_fast_pitch_util_pitch_s24, 219
ags_fast_pitch_util_pitch_s32, 219
ags_fast_pitch_util_pitch_s64, 219
ags_fast_pitch_util_pitch_s8, 218
ags_fast_pitch_util_set_base_key, 217
ags_fast_pitch_util_set_buffer_length, 215
ags_fast_pitch_util_set_destination, 214
ags_fast_pitch_util_set_destination_stride, 215
ags_fast_pitch_util_set_format, 216
ags_fast_pitch_util_set_samplerate, 217
ags_fast_pitch_util_set_source, 213
ags_fast_pitch_util_set_source_stride, 214
ags_fast_pitch_util_set_tuning, 218
AgsFastPitchUtil, 221
AGS_FIFOOUT, 640
AgsFifoout, 642
AgsFifoout:attack, 642
AgsFifoout:bpm, 643
AgsFifoout:buffer, 643
AgsFifoout:buffer-size, 643
AgsFifoout:delay-factor, 643

AgsFifoout:device, 644
AgsFifoout:dsp-channels, 644
AgsFifoout:format, 644
AgsFifoout:pcm-channels, 644
AgsFifoout:samplerate, 645
ags_fifoout_adjust_delay_and_attack, 639
AGS_FIFOOUT_CLASS, 640
AGS_FIFOOUT_DEFAULT_APP_BUFFER_SIZE, 640
AGS_FIFOOUT_DEFAULT_DEVICE, 640
AGS_FIFOOUT_DEFAULT_RING_BUFFER_SIZE, 641
AGS_FIFOOUT_ERROR, 642
ags_fifoout_error_quark, 637
ags_fifoout_flags_get_type, 640
AGS_FIFOOUT_GET_CLASS, 640
AGS_FIFOOUT_GET_OBJ_MUTEX, 637
ags_fifoout_get_type, 640
ags_fifoout_new, 639
ags_fifoout_realloc_buffer, 639
ags_fifoout_set_flags, 638
ags_fifoout_switch_buffer_flag, 639
ags_fifoout_test_flags, 638
ags_fifoout_unset_flags, 638
AgsFifooutAppBufferMode, 641
AgsFifooutClass, 642
AgsFifooutError, 642
AgsFifooutFlags, 641
AGS_FLUID_CENTS_HZ_SIZE, 554
ags_fluid_ct2hz, 553
ags_fluid_ct2hz_real, 553
AGS_FLUID_FRACT_MAX, 554
ags_fluid_iir_filter_type_get_type, 619
AGS_FLUID_IIR_FILTER_UTIL, 619
ags_fluid_iir_filter_util_alloc, 608
ags_fluid_iir_filter_util_copy, 608
ags_fluid_iir_filter_util_free, 608
ags_fluid_iir_filter_util_get_buffer_length, 611
ags_fluid_iir_filter_util_get_destination, 610
ags_fluid_iir_filter_util_get_destination_stride, 610
ags_fluid_iir_filter_util_get_filter_gain, 616
ags_fluid_iir_filter_util_get_filter_startup, 614
ags_fluid_iir_filter_util_get_filter_type, 613
ags_fluid_iir_filter_util_get_flags, 614
ags_fluid_iir_filter_util_get_format, 612
ags_fluid_iir_filter_util_get_q_lin, 615
ags_fluid_iir_filter_util_get_samplerate, 612
ags_fluid_iir_filter_util_get_source, 608
ags_fluid_iir_filter_util_get_source_stride, 609
ags_fluid_iir_filter_util_get_type, 619
ags_fluid_iir_filter_util_process, 619
ags_fluid_iir_filter_util_process_complex, 618
ags_fluid_iir_filter_util_process_double, 618
ags_fluid_iir_filter_util_process_float, 618
ags_fluid_iir_filter_util_process_s16, 617
ags_fluid_iir_filter_util_process_s24, 617
ags_fluid_iir_filter_util_process_s32, 617
ags_fluid_iir_filter_util_process_s64, 617
ags_fluid_iir_filter_util_process_s8, 616
ags_fluid_iir_filter_util_set_buffer_length, 611
ags_fluid_iir_filter_util_set_destination, 610
ags_fluid_iir_filter_util_set_destination_stride, 611
ags_fluid_iir_filter_util_set_filter_gain, 616
ags_fluid_iir_filter_util_set_filter_startup, 615
ags_fluid_iir_filter_util_set_filter_type, 613
ags_fluid_iir_filter_util_set_flags, 614
ags_fluid_iir_filter_util_set_format, 612
ags_fluid_iir_filter_util_set_q_lin, 615
ags_fluid_iir_filter_util_set_samplerate, 613
ags_fluid_iir_filter_util_set_source, 609
ags_fluid_iir_filter_util_set_source_stride, 609
AGS_FLUID_INTERP_BITS, 554
AGS_FLUID_INTERP_BITS_MASK, 554
AGS_FLUID_INTERP_BITS_SHIFT, 554
AGS_FLUID_INTERP_MAX, 554
AGS_FLUID_INTERPOLATE_4TH_ORDER_UTIL, 592
ags_fluid_interpolate_4th_order_util_alloc, 581
ags_fluid_interpolate_4th_order_util_copy, 581
ags_fluid_interpolate_4th_order_util_free, 582
ags_fluid_interpolate_4th_order_util_get_base_key, 587
ags_fluid_interpolate_4th_order_util_get_buffer_length, 585
ags_fluid_interpolate_4th_order_util_get_destination, 583
ags_fluid_interpolate_4th_order_util_get_destination_stride, 584
ags_fluid_interpolate_4th_order_util_get_format, 586
ags_fluid_interpolate_4th_order_util_get_phase_increment, 589
ags_fluid_interpolate_4th_order_util_get_samplerate, 586
ags_fluid_interpolate_4th_order_util_get_source, 582
ags_fluid_interpolate_4th_order_util_get_source_stride, 583
ags_fluid_interpolate_4th_order_util_get_tuning, 588
ags_fluid_interpolate_4th_order_util_get_type, 592
ags_fluid_interpolate_4th_order_util_pitch, 592
ags_fluid_interpolate_4th_order_util_pitch_complex, 592
ags_fluid_interpolate_4th_order_util_pitch_double, 591
ags_fluid_interpolate_4th_order_util_pitch_float, 591
ags_fluid_interpolate_4th_order_util_pitch_s16, 590
ags_fluid_interpolate_4th_order_util_pitch_s24, 590
ags_fluid_interpolate_4th_order_util_pitch_s32, 590
ags_fluid_interpolate_4th_order_util_pitch_s64, 591
ags_fluid_interpolate_4th_order_util_pitch_s8, 589
ags_fluid_interpolate_4th_order_util_set_base_key, 588
ags_fluid_interpolate_4th_order_util_set_buffer_length, 585
ags_fluid_interpolate_4th_order_util_set_destination, 584
ags_fluid_interpolate_4th_order_util_set_destination_stride, 585
ags_fluid_interpolate_4th_order_util_set_format, 586
ags_fluid_interpolate_4th_order_util_set_phase_increment, 589
ags_fluid_interpolate_4th_order_util_set_samplerate, 587
ags_fluid_interpolate_4th_order_util_set_source, 582
ags_fluid_interpolate_4th_order_util_set_source_stride, 583
ags_fluid_interpolate_4th_order_util_set_tuning, 588
AGS_FLUID_INTERPOLATE_7TH_ORDER_UTIL, 605
ags_fluid_interpolate_7th_order_util_alloc, 594
ags_fluid_interpolate_7th_order_util_copy, 595
ags_fluid_interpolate_7th_order_util_free, 595

ags_fluid_interpolate_7th_order_util_get_base_key, 600
ags_fluid_interpolate_7th_order_util_get_buffer_length, 598
ags_fluid_interpolate_7th_order_util_get_destination, 597
ags_fluid_interpolate_7th_order_util_get_destination_stride, 597
ags_fluid_interpolate_7th_order_util_get_format, 599
ags_fluid_interpolate_7th_order_util_get_phase_increment, 602
ags_fluid_interpolate_7th_order_util_get_samplerate, 600
ags_fluid_interpolate_7th_order_util_get_source, 595
ags_fluid_interpolate_7th_order_util_get_source_stride, 596
ags_fluid_interpolate_7th_order_util_get_tuning, 601
ags_fluid_interpolate_7th_order_util_get_type, 606
ags_fluid_interpolate_7th_order_util_pitch, 605
ags_fluid_interpolate_7th_order_util_pitch_complex, 605
ags_fluid_interpolate_7th_order_util_pitch_double, 605
ags_fluid_interpolate_7th_order_util_pitch_float, 604
ags_fluid_interpolate_7th_order_util_pitch_s16, 603
ags_fluid_interpolate_7th_order_util_pitch_s24, 603
ags_fluid_interpolate_7th_order_util_pitch_s32, 604
ags_fluid_interpolate_7th_order_util_pitch_s64, 604
ags_fluid_interpolate_7th_order_util_pitch_s8, 603
ags_fluid_interpolate_7th_order_util_set_base_key, 601
ags_fluid_interpolate_7th_order_util_set_buffer_length, 599
ags_fluid_interpolate_7th_order_util_set_destination, 597
ags_fluid_interpolate_7th_order_util_set_destination_stride, 598
ags_fluid_interpolate_7th_order_util_set_format, 599
ags_fluid_interpolate_7th_order_util_set_phase_increment, 602
ags_fluid_interpolate_7th_order_util_set_samplerate, 600
ags_fluid_interpolate_7th_order_util_set_source, 596
ags_fluid_interpolate_7th_order_util_set_source_stride, 596
ags_fluid_interpolate_7th_order_util_set_tuning, 602
AGS_FLUID_INTERPOLATE_LINEAR_UTIL, 579
ags_fluid_interpolate_linear_util_alloc, 568
ags_fluid_interpolate_linear_util_copy, 568
ags_fluid_interpolate_linear_util_free, 569
ags_fluid_interpolate_linear_util_get_base_key, 574
ags_fluid_interpolate_linear_util_get_buffer_length, 572
ags_fluid_interpolate_linear_util_get_destination, 570
ags_fluid_interpolate_linear_util_get_destination_stride, 571
ags_fluid_interpolate_linear_util_get_format, 572
ags_fluid_interpolate_linear_util_get_phase_increment, 575
ags_fluid_interpolate_linear_util_get_samplerate, 573
ags_fluid_interpolate_linear_util_get_source, 569
ags_fluid_interpolate_linear_util_get_source_stride, 570
ags_fluid_interpolate_linear_util_get_tuning, 575
ags_fluid_interpolate_linear_util_get_type, 579
ags_fluid_interpolate_linear_util_pitch, 579
ags_fluid_interpolate_linear_util_pitch_complex, 578
ags_fluid_interpolate_linear_util_pitch_double, 578
ags_fluid_interpolate_linear_util_pitch_float, 578
ags_fluid_interpolate_linear_util_pitch_s16, 576
ags_fluid_interpolate_linear_util_pitch_s24, 577
ags_fluid_interpolate_linear_util_pitch_s32, 577
ags_fluid_interpolate_linear_util_pitch_s64, 577
ags_fluid_interpolate_linear_util_pitch_s8, 576
ags_fluid_interpolate_linear_util_set_base_key, 574
ags_fluid_interpolate_linear_util_set_buffer_length, 572
ags_fluid_interpolate_linear_util_set_destination, 571
ags_fluid_interpolate_linear_util_set_destination_stride, 571
ags_fluid_interpolate_linear_util_set_format, 573
ags_fluid_interpolate_linear_util_set_phase_increment, 576
ags_fluid_interpolate_linear_util_set_samplerate, 574
ags_fluid_interpolate_linear_util_set_source, 569
ags_fluid_interpolate_linear_util_set_source_stride, 570
ags_fluid_interpolate_linear_util_set_tuning, 575
AGS_FLUID_INTERPOLATE_NONE_UTIL, 566
ags_fluid_interpolate_none_util_alloc, 556
ags_fluid_interpolate_none_util_copy, 556
ags_fluid_interpolate_none_util_free, 556
ags_fluid_interpolate_none_util_get_base_key, 561
ags_fluid_interpolate_none_util_get_buffer_length, 559
ags_fluid_interpolate_none_util_get_destination, 558
ags_fluid_interpolate_none_util_get_destination_stride, 559
ags_fluid_interpolate_none_util_get_format, 560
ags_fluid_interpolate_none_util_get_phase_increment, 563
ags_fluid_interpolate_none_util_get_samplerate, 561
ags_fluid_interpolate_none_util_get_source, 557
ags_fluid_interpolate_none_util_get_source_stride, 557
ags_fluid_interpolate_none_util_get_tuning, 562
ags_fluid_interpolate_none_util_get_type, 566
ags_fluid_interpolate_none_util_pitch, 566
ags_fluid_interpolate_none_util_pitch_complex, 565
ags_fluid_interpolate_none_util_pitch_double, 565
ags_fluid_interpolate_none_util_pitch_float, 565
ags_fluid_interpolate_none_util_pitch_s16, 564
ags_fluid_interpolate_none_util_pitch_s24, 564
ags_fluid_interpolate_none_util_pitch_s32, 564
ags_fluid_interpolate_none_util_pitch_s64, 565
ags_fluid_interpolate_none_util_pitch_s8, 563
ags_fluid_interpolate_none_util_set_base_key, 562
ags_fluid_interpolate_none_util_set_buffer_length, 560
ags_fluid_interpolate_none_util_set_destination, 558
ags_fluid_interpolate_none_util_set_destination_stride, 559
ags_fluid_interpolate_none_util_set_format, 560
ags_fluid_interpolate_none_util_set_phase_increment, 563
ags_fluid_interpolate_none_util_set_samplerate, 561
ags_fluid_interpolate_none_util_set_source, 557
ags_fluid_interpolate_none_util_set_source_stride, 558
ags_fluid_interpolate_none_util_set_tuning, 562
ags_fluid_phase_decr, 553
ags_fluid_phase_double, 553
ags_fluid_phase_fract, 552
ags_fluid_phase_fract_to_tablerow, 553
ags_fluid_phase_from_index_fract, 552
ags_fluid_phase_incr, 553
ags_fluid_phase_index, 552
ags_fluid_phase_index_plusplus, 553
ags_fluid_phase_index_round, 553
ags_fluid_phase_set, 552
ags_fluid_phase_set_float, 552
ags_fluid_phase_set_int, 552
ags_fluid_phase_sub_int, 553

AGS_FLUID_SINC_INTERP_ORDER, 554
ags_fluid_util_get_type, 553
AgsFluidIIRFilterType, 619
AgsFluidIIRFilterUtil, 620
AgsFluidInterpolate4thOrderUtil, 593
AgsFluidInterpolate7thOrderUtil, 606
AgsFluidInterpolateLinearUtil, 579
AgsFluidInterpolateNoneUtil, 566
AgsFluidUtil, 554
ags_fm_synth_util_alloc, 403
ags_fm_synth_util_compute_impulse, 426
ags_fm_synth_util_compute_impulse_complex, 426
ags_fm_synth_util_compute_impulse_double, 425
ags_fm_synth_util_compute_impulse_float, 425
ags_fm_synth_util_compute_impulse_s16, 424
ags_fm_synth_util_compute_impulse_s24, 424
ags_fm_synth_util_compute_impulse_s32, 424
ags_fm_synth_util_compute_impulse_s64, 425
ags_fm_synth_util_compute_impulse_s8, 424
ags_fm_synth_util_compute_sawtooth, 418
ags_fm_synth_util_compute_sawtooth_complex, 418
ags_fm_synth_util_compute_sawtooth_double, 418
ags_fm_synth_util_compute_sawtooth_float, 418
ags_fm_synth_util_compute_sawtooth_s16, 416
ags_fm_synth_util_compute_sawtooth_s24, 417
ags_fm_synth_util_compute_sawtooth_s32, 417
ags_fm_synth_util_compute_sawtooth_s64, 417
ags_fm_synth_util_compute_sawtooth_s8, 416
ags_fm_synth_util_compute_sin, 416
ags_fm_synth_util_compute_sin_complex, 416
ags_fm_synth_util_compute_sin_double, 415
ags_fm_synth_util_compute_sin_float, 415
ags_fm_synth_util_compute_sin_s16, 414
ags_fm_synth_util_compute_sin_s24, 414
ags_fm_synth_util_compute_sin_s32, 415
ags_fm_synth_util_compute_sin_s64, 415
ags_fm_synth_util_compute_sin_s8, 414
ags_fm_synth_util_compute_square, 423
ags_fm_synth_util_compute_square_complex, 423
ags_fm_synth_util_compute_square_double, 423
ags_fm_synth_util_compute_square_float, 423
ags_fm_synth_util_compute_square_s16, 422
ags_fm_synth_util_compute_square_s24, 422
ags_fm_synth_util_compute_square_s32, 422
ags_fm_synth_util_compute_square_s64, 422
ags_fm_synth_util_compute_square_s8, 421
ags_fm_synth_util_compute_triangle, 421
ags_fm_synth_util_compute_triangle_complex, 421
ags_fm_synth_util_compute_triangle_double, 420
ags_fm_synth_util_compute_triangle_float, 420
ags_fm_synth_util_compute_triangle_s16, 419
ags_fm_synth_util_compute_triangle_s24, 419
ags_fm_synth_util_compute_triangle_s32, 420
ags_fm_synth_util_compute_triangle_s64, 420
ags_fm_synth_util_compute_triangle_s8, 419
ags_fm_synth_util_copy, 404
AGS_FM_SYNTH_UTIL_DEFAULT_FREQUENCY, 426
AGS_FM_SYNTH_UTIL_DEFAULT_LFO_DEPTH, 426
AGS_FM_SYNTH_UTIL_DEFAULT_LFO_FREQUENCY, 426
AGS_FM_SYNTH_UTIL_DEFAULT_TUNING, 427
ags_fm_synth_util_free, 404
ags_fm_synth_util_get_buffer_length, 405
ags_fm_synth_util_get_format, 406
ags_fm_synth_util_get_frame_count, 413
ags_fm_synth_util_get_frequency, 408
ags_fm_synth_util_get_lfo_depth, 411
ags_fm_synth_util_get_lfo_frequency, 411
ags_fm_synth_util_get_lfo_oscillator_mode, 410
ags_fm_synth_util_get_offset, 413
ags_fm_synth_util_get_phase, 409
ags_fm_synth_util_get_samplerate, 407
ags_fm_synth_util_get_source, 404
ags_fm_synth_util_get_source_stride, 405
ags_fm_synth_util_get_synth_oscillator_mode, 407
ags_fm_synth_util_get_tuning, 412
ags_fm_synth_util_get_type, 426
ags_fm_synth_util_get_volume, 409
ags_fm_synth_util_set_buffer_length, 406
ags_fm_synth_util_set_format, 406
ags_fm_synth_util_set_frame_count, 413
ags_fm_synth_util_set_frequency, 408
ags_fm_synth_util_set_lfo_depth, 412
ags_fm_synth_util_set_lfo_frequency, 411
ags_fm_synth_util_set_lfo_oscillator_mode, 410
ags_fm_synth_util_set_offset, 414
ags_fm_synth_util_set_phase, 409
ags_fm_synth_util_set_samplerate, 407
ags_fm_synth_util_set_source, 405
ags_fm_synth_util_set_source_stride, 405
ags_fm_synth_util_set_synth_oscillator_mode, 408
ags_fm_synth_util_set_tuning, 412
ags_fm_synth_util_set_volume, 410
AgsFMSynthUtil, 427
AGS_FOURIER_TRANSFORM_UTIL_ANALYSIS_WINDOW, 204
ags_fourier_transform_util_compute_stft_complex, 207
ags_fourier_transform_util_compute_stft_double, 206
AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_DOU 203
ags_fourier_transform_util_compute_stft_float, 206
AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_FLO 203
ags_fourier_transform_util_compute_stft_s16, 204
AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S16_ 202
ags_fourier_transform_util_compute_stft_s24, 205
AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S24_ 202
ags_fourier_transform_util_compute_stft_s32, 205
AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S32_ 202
ags_fourier_transform_util_compute_stft_s64, 205

AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S64_FRAME, 203
ags_fourier_transform_util_compute_stft_s8, 204
AGS_FOURIER_TRANSFORM_UTIL_COMPUTE_STFT_S8_FRAME, 202
ags_fourier_transform_util_get_type, 210
ags_fourier_transform_util_inverse_stft_complex, 210
ags_fourier_transform_util_inverse_stft_double, 209
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_DOUBLE_FRAME, 204
ags_fourier_transform_util_inverse_stft_float, 209
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_FLOAT_FRAME, 203
ags_fourier_transform_util_inverse_stft_s16, 207
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S16_FRAME, 203
ags_fourier_transform_util_inverse_stft_s24, 208
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S24_FRAME, 203
ags_fourier_transform_util_inverse_stft_s32, 208
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S32_FRAME, 203
ags_fourier_transform_util_inverse_stft_s64, 209
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S64_FRAME, 203
ags_fourier_transform_util_inverse_stft_s8, 207
AGS_FOURIER_TRANSFORM_UTIL_INVERSE_STFT_S8_FRAME, 203
AgsFourierTransformUtil, 210
AGS_FREE_SELECTION, 2335
AGS_FREE_SELECTION_CLASS, 2335
AGS_FREE_SELECTION_GET_CLASS, 2335
ags_free_selection_get_type, 2336
ags_free_selection_new, 2335
AgsFreeSelection, 2336
AgsFreeSelection:notation, 2336
AgsFreeSelectionClass, 2336
ags_frequency_aliase_util_alloc, 257
ags_frequency_aliase_util_copy, 258
AGS_FREQUENCY_ALIASE_UTIL_DEFAULT_FORMAT, 266
ags_frequency_aliase_util_free, 258
ags_frequency_aliase_util_get_buffer_length, 262
ags_frequency_aliase_util_get_destination, 258
ags_frequency_aliase_util_get_destination_stride, 259
ags_frequency_aliase_util_get_format, 263
ags_frequency_aliase_util_get_phase_shifted_source, 261
ags_frequency_aliase_util_get_phase_shifted_source_stride, 262
ags_frequency_aliase_util_get_source, 260
ags_frequency_aliase_util_get_source_stride, 260
ags_frequency_aliase_util_get_type, 266
ags_frequency_aliase_util_process, 266
ags_frequency_aliase_util_process_complex, 265
ags_frequency_aliase_util_process_double, 265
ags_frequency_aliase_util_process_float, 265
ags_frequency_aliase_util_process_s16, 264
ags_frequency_aliase_util_process_s24, 264
ags_frequency_aliase_util_process_s32, 264
ags_frequency_aliase_util_process_s64, 265
ags_frequency_aliase_util_process_s8, 263
ags_frequency_aliase_util_set_buffer_length, 262
ags_frequency_aliase_util_set_destination, 259
ags_frequency_aliase_util_set_destination_stride, 259
ags_frequency_aliase_util_set_format, 263
ags_frequency_aliase_util_set_phase_shifted_source, 261
ags_frequency_aliase_util_set_phase_shifted_source_stride, 262
ags_frequency_aliase_util_set_source, 260
ags_frequency_aliase_util_set_source_stride, 261
AGS_FREQUENCY_MAP, 631
AGS_FREQUENCY_MAP_CLASS, 631
ags_frequency_map_compute_max_likelihood, 630
AGS_FREQUENCY_MAP_DEFAULT_FREQ, 632
ags_frequency_map_factorize, 630
AGS_FREQUENCY_MAP_GET_CLASS, 631
AGS_FREQUENCY_MAP_GET_OBJ_MUTEX, 628
ags_frequency_map_get_type, 632
AGS_FREQUENCY_MAP_MANAGER, 626
ags_frequency_map_manager_add_factorized_frequency_map, 623
ags_frequency_map_manager_add_frequency_map, 623
AGS_FREQUENCY_MAP_MANAGER_CLASS, 626
AGS_FREQUENCY_MAP_MANAGER_EQUINOX, 626
ags_frequency_map_manager_find_factorized_frequency_map, 624
ags_frequency_map_manager_find_frequency_map, 624
AGS_FREQUENCY_MAP_MANAGER_GET_CLASS, 626
ags_frequency_map_manager_get_instance, 625
AGS_FREQUENCY_MAP_MANAGER_GET_OBJ_MUTEX, 622
ags_frequency_map_manager_get_type, 626
ags_frequency_map_manager_load_default, 625
ags_frequency_map_manager_new, 625
ags_frequency_map_manager_set_flags, 622
ags_frequency_map_manager_test_flags, 622
ags_frequency_map_manager_unset_flags, 623
ags_frequency_map_new, 631
ags_frequency_map_process, 630
ags_frequency_map_set_flags, 629
ags_frequency_map_sort_func, 629
ags_frequency_map_test_flags, 629
ags_frequency_map_unset_flags, 630
AgsFrequencyAliaseUtil, 266
AgsFrequencyMap, 632
AgsFrequencyMap::compute-max-likelihood, 634
AgsFrequencyMap::factorize, 635
AgsFrequencyMap::process, 635
AgsFrequencyMap:attack, 633
AgsFrequencyMap:buffer-size, 633
AgsFrequencyMap:frame-count, 633
AgsFrequencyMap:freq, 633
AgsFrequencyMap:oscillator-mode, 634
AgsFrequencyMap:samplerate, 634

AgsFrequencyMap:window-count, 634
AgsFrequencyMap:z-index, 634
AgsFrequencyMapClass, 632
AgsFrequencyMapFlags, 632
AgsFrequencyMapManager, 627
AgsFrequencyMapManagerClass, 627
AgsFrequencyMapManagerFlags, 626
AGS_FX_ANALYSE_AUDIO, 2123
AGS_FX_ANALYSE_AUDIO_CLASS, 2123
AGS_FX_ANALYSE_AUDIO_GET_CLASS, 2124
ags_fx_analyse_audio_get_type, 2124
ags_fx_analyse_audio_new, 2123
AGS_FX_ANALYSE_AUDIO_PROCESSOR, 2126
AGS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS, 2126
AGS_FX_ANALYSE_AUDIO_PROCESSOR_GET_CLASS, 2126
ags_fx_analyse_audio_processor_get_type, 2126
ags_fx_analyse_audio_processor_new, 2125
AGS_FX_ANALYSE_AUDIO_SIGNAL, 2128
AGS_FX_ANALYSE_AUDIO_SIGNAL_CLASS, 2128
AGS_FX_ANALYSE_AUDIO_SIGNAL_GET_CLASS, 2128
ags_fx_analyse_audio_signal_get_type, 2128
ags_fx_analyse_audio_signal_new, 2128
AGS_FX_ANALYSE_CHANNEL, 2133
AGS_FX_ANALYSE_CHANNEL_CLASS, 2133
AGS_FX_ANALYSE_CHANNEL_GET_CLASS, 2133
ags_fx_analyse_channel_get_input_data, 2132
ags_fx_analyse_channel_get_type, 2134
AGS_FX_ANALYSE_CHANNEL_INPUT_DATA, 2130
ags_fx_analyse_channel_input_data_alloc, 2130
ags_fx_analyse_channel_input_data_free, 2131
AGS_FX_ANALYSE_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX, 2130
ags_fx_analyse_channel_input_data_get_strct_mutex, 2131
ags_fx_analyse_channel_input_get_in, 2132
ags_fx_analyse_channel_input_get_out, 2132
ags_fx_analyse_channel_input_get_parent, 2131
ags_fx_analyse_channel_new, 2133
AGS_FX_ANALYSE_CHANNEL_PROCESSOR, 2136
AGS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS, 2136
AGS_FX_ANALYSE_CHANNEL_PROCESSOR_GET_CLASS, 2136
ags_fx_analyse_channel_processor_get_type, 2137
ags_fx_analyse_channel_processor_new, 2136
AGS_FX_ANALYSE_RECYCLING, 2138
AGS_FX_ANALYSE_RECYCLING_CLASS, 2138
AGS_FX_ANALYSE_RECYCLING_GET_CLASS, 2139
ags_fx_analyse_recycling_get_type, 2139
ags_fx_analyse_recycling_new, 2138
AGS_FX_BUFFER_AUDIO, 1936
AGS_FX_BUFFER_AUDIO_CLASS, 1936
AGS_FX_BUFFER_AUDIO_GET_CLASS, 1936
ags_fx_buffer_audio_get_type, 1936
ags_fx_buffer_audio_new, 1936
AGS_FX_BUFFER_AUDIO_PROCESSOR, 1938
AGS_FX_BUFFER_AUDIO_PROCESSOR_CLASS, 1938
AGS_FX_BUFFER_AUDIO_PROCESSOR_GET_CLASS, 1938
ags_fx_buffer_audio_processor_get_type, 1939
ags_fx_buffer_audio_processor_new, 1938
AGS_FX_BUFFER_AUDIO_SIGNAL, 1941
AGS_FX_BUFFER_AUDIO_SIGNAL_CLASS, 1941
AGS_FX_BUFFER_AUDIO_SIGNAL_GET_CLASS, 1941
ags_fx_buffer_audio_signal_get_type, 1941
ags_fx_buffer_audio_signal_new, 1940
AGS_FX_BUFFER_CHANNEL, 1945
AGS_FX_BUFFER_CHANNEL_CLASS, 1946
AGS_FX_BUFFER_CHANNEL_GET_CLASS, 1946
ags_fx_buffer_channel_get_input_data, 1945
ags_fx_buffer_channel_get_type, 1946
AGS_FX_BUFFER_CHANNEL_INPUT_DATA, 1943
ags_fx_buffer_channel_input_data_alloc, 1943
ags_fx_buffer_channel_input_data_free, 1943
AGS_FX_BUFFER_CHANNEL_INPUT_DATA_GET_STRCT_M 1943
ags_fx_buffer_channel_input_data_get_strct_mutex, 1944
ags_fx_buffer_channel_input_get_destination, 1944
ags_fx_buffer_channel_input_get_parent, 1944
ags_fx_buffer_channel_new, 1945
AGS_FX_BUFFER_CHANNEL_PROCESSOR, 1948
AGS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS, 1948
AGS_FX_BUFFER_CHANNEL_PROCESSOR_GET_CLASS, 1948
ags_fx_buffer_channel_processor_get_type, 1948
ags_fx_buffer_channel_processor_new, 1948
AGS_FX_BUFFER_RECYCLING, 1950
AGS_FX_BUFFER_RECYCLING_CLASS, 1950
AGS_FX_BUFFER_RECYCLING_GET_CLASS, 1950
ags_fx_buffer_recycling_get_type, 1951
ags_fx_buffer_recycling_new, 1950
AGS_FX_DSSI_AUDIO, 1998
ags_fx_dssi_audio_change_program, 1997
AGS_FX_DSSI_AUDIO_CHANNEL_DATA, 1993
ags_fx_dssi_audio_channel_data_alloc, 1994
ags_fx_dssi_audio_channel_data_free, 1995
AGS_FX_DSSI_AUDIO_CHANNEL_DATA_GET_STRCT_MU 1993
AGS_FX_DSSI_AUDIO_CLASS, 1998
AGS_FX_DSSI_AUDIO_GET_CLASS, 1998
ags_fx_dssi_audio_get_type, 1998
AGS_FX_DSSI_AUDIO_INPUT_DATA, 1994
ags_fx_dssi_audio_input_data_alloc, 1995
ags_fx_dssi_audio_input_data_free, 1995
AGS_FX_DSSI_AUDIO_INPUT_DATA_GET_STRCT_MUTEX 1994
ags_fx_dssi_audio_load_plugin, 1996
ags_fx_dssi_audio_load_port, 1997
ags_fx_dssi_audio_new, 1997
AGS_FX_DSSI_AUDIO_PROCESSOR, 2001
AGS_FX_DSSI_AUDIO_PROCESSOR_CLASS, 2001
AGS_FX_DSSI_AUDIO_PROCESSOR_GET_CLASS, 2001
ags_fx_dssi_audio_processor_get_type, 2001
ags_fx_dssi_audio_processor_new, 2000

AGS_FX_DSSI_AUDIO_SCOPE_DATA, 1993
ags_fx_dssi_audio_scope_data_alloc, 1994
ags_fx_dssi_audio_scope_data_free, 1994
AGS_FX_DSSI_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX, 1993
ags_fx_dssi_audio_set_flags, 1996
AGS_FX_DSSI_AUDIO_SIGNAL, 2003
AGS_FX_DSSI_AUDIO_SIGNAL_CLASS, 2003
AGS_FX_DSSI_AUDIO_SIGNAL_GET_CLASS, 2003
ags_fx_dssi_audio_signal_get_type, 2004
ags_fx_dssi_audio_signal_new, 2003
ags_fx_dssi_audio_test_flags, 1995
ags_fx_dssi_audio_unset_flags, 1996
AGS_FX_DSSI_CHANNEL, 2006
AGS_FX_DSSI_CHANNEL_CLASS, 2006
AGS_FX_DSSI_CHANNEL_GET_CLASS, 2006
ags_fx_dssi_channel_get_type, 2006
ags_fx_dssi_channel_load_port, 2005
ags_fx_dssi_channel_new, 2005
AGS_FX_DSSI_CHANNEL_PROCESSOR, 2008
AGS_FX_DSSI_CHANNEL_PROCESSOR_CLASS, 2008
AGS_FX_DSSI_CHANNEL_PROCESSOR_GET_CLASS, 2008
ags_fx_dssi_channel_processor_get_type, 2009
ags_fx_dssi_channel_processor_new, 2008
AGS_FX_DSSI_RECYCLING, 2010
AGS_FX_DSSI_RECYCLING_CLASS, 2011
AGS_FX_DSSI_RECYCLING_GET_CLASS, 2011
ags_fx_dssi_recycling_get_type, 2011
ags_fx_dssi_recycling_new, 2010
AGS_FX_ENVELOPE_AUDIO, 2203
AGS_FX_ENVELOPE_AUDIO_CLASS, 2203
AGS_FX_ENVELOPE_AUDIO_GET_CLASS, 2203
ags_fx_envelope_audio_get_type, 2203
ags_fx_envelope_audio_new, 2203
AGS_FX_ENVELOPE_AUDIO_PROCESSOR, 2205
AGS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS, 2205
AGS_FX_ENVELOPE_AUDIO_PROCESSOR_GET_CLASS, 2205
ags_fx_envelope_audio_processor_get_type, 2206
ags_fx_envelope_audio_processor_new, 2205
AGS_FX_ENVELOPE_AUDIO_SIGNAL, 2208
AGS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS, 2208
AGS_FX_ENVELOPE_AUDIO_SIGNAL_GET_CLASS, 2208
ags_fx_envelope_audio_signal_get_type, 2208
ags_fx_envelope_audio_signal_new, 2207
AGS_FX_ENVELOPE_CHANNEL, 2210
AGS_FX_ENVELOPE_CHANNEL_CLASS, 2210
AGS_FX_ENVELOPE_CHANNEL_GET_CLASS, 2210
ags_fx_envelope_channel_get_type, 2211
ags_fx_envelope_channel_new, 2210
AGS_FX_ENVELOPE_CHANNEL_PROCESSOR, 2214
AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS, 2214
AGS_FX_ENVELOPE_CHANNEL_PROCESSOR_GET_CLASS, 2214
ags_fx_envelope_channel_processor_get_type, 2214
ags_fx_envelope_channel_processor_new, 2213
AGS_FX_ENVELOPE_RECYCLING, 2216
AGS_FX_ENVELOPE_RECYCLING_CLASS, 2216
AGS_FX_ENVELOPE_RECYCLING_GET_CLASS, 2216
ags_fx_envelope_recycling_get_type, 2217
ags_fx_envelope_recycling_new, 2216
AGS_FX_EQ10_AUDIO, 2185
AGS_FX_EQ10_AUDIO_CLASS, 2185
AGS_FX_EQ10_AUDIO_GET_CLASS, 2185
ags_fx_eq10_audio_get_type, 2186
ags_fx_eq10_audio_new, 2185
AGS_FX_EQ10_AUDIO_PROCESSOR, 2188
AGS_FX_EQ10_AUDIO_PROCESSOR_CLASS, 2188
AGS_FX_EQ10_AUDIO_PROCESSOR_GET_CLASS, 2188
ags_fx_eq10_audio_processor_get_type, 2188
ags_fx_eq10_audio_processor_new, 2187
AGS_FX_EQ10_AUDIO_SIGNAL, 2190
AGS_FX_EQ10_AUDIO_SIGNAL_CLASS, 2190
AGS_FX_EQ10_AUDIO_SIGNAL_GET_CLASS, 2190
ags_fx_eq10_audio_signal_get_type, 2190
ags_fx_eq10_audio_signal_new, 2190
AGS_FX_EQ10_CHANNEL, 2193
AGS_FX_EQ10_CHANNEL_CLASS, 2193
AGS_FX_EQ10_CHANNEL_GET_CLASS, 2194
ags_fx_eq10_channel_get_type, 2194
AGS_FX_EQ10_CHANNEL_INPUT_DATA, 2192
ags_fx_eq10_channel_input_data_alloc, 2192
AGS_FX_EQ10_CHANNEL_INPUT_DATA_CACHE_SIZE, 2194
ags_fx_eq10_channel_input_data_free, 2193
AGS_FX_EQ10_CHANNEL_INPUT_DATA_GET_STRCT_MUT 2192
ags_fx_eq10_channel_new, 2193
AGS_FX_EQ10_CHANNEL_PROCESSOR, 2198
AGS_FX_EQ10_CHANNEL_PROCESSOR_CLASS, 2198
AGS_FX_EQ10_CHANNEL_PROCESSOR_GET_CLASS, 2198
ags_fx_eq10_channel_processor_get_type, 2199
ags_fx_eq10_channel_processor_new, 2198
AGS_FX_EQ10_RECYCLING, 2201
AGS_FX_EQ10_RECYCLING_CLASS, 2201
AGS_FX_EQ10_RECYCLING_GET_CLASS, 2201
ags_fx_eq10_recycling_get_type, 2201
ags_fx_eq10_recycling_new, 2200
AGS_FX_FACTORY, 1320
AGS_FX_FACTORY_CLASS, 1320
ags_fx_factory_create, 1319
ags_fx_factory_create_flags_get_type, 1320
AGS_FX_FACTORY_GET_CLASS, 1320
ags_fx_factory_get_type, 1320
AGS_FX_FM_SYNTH_AUDIO, 2064
ags_fx_fm_synth_audio_channel_data_alloc, 2062
ags_fx_fm_synth_audio_channel_data_free, 2063
AGS_FX_FM_SYNTH_AUDIO_CLASS, 2064
AGS_FX_FM_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE, 2065
AGS_FX_FM_SYNTH_AUDIO_GET_CLASS, 2064

ags_fx_fm_synth_audio_get_type, 2064
ags_fx_fm_synth_audio_input_data_alloc, 2063
ags_fx_fm_synth_audio_input_data_free, 2063
ags_fx_fm_synth_audio_new, 2064
AGS_FX_FM_SYNTH_AUDIO_PROCESSOR, 2076
AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS, 2077
AGS_FX_FM_SYNTH_AUDIO_PROCESSOR_GET_CLASS, 2077
ags_fx_fm_synth_audio_processor_get_type, 2077
ags_fx_fm_synth_audio_processor_new, 2076
ags_fx_fm_synth_audio_scope_data_alloc, 2062
ags_fx_fm_synth_audio_scope_data_free, 2062
AGS_FX_FM_SYNTH_AUDIO_SIGNAL, 2079
AGS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS, 2079
AGS_FX_FM_SYNTH_AUDIO_SIGNAL_GET_CLASS, 2079
ags_fx_fm_synth_audio_signal_get_type, 2079
ags_fx_fm_synth_audio_signal_new, 2078
AGS_FX_FM_SYNTH_CHANNEL, 2081
AGS_FX_FM_SYNTH_CHANNEL_CLASS, 2081
AGS_FX_FM_SYNTH_CHANNEL_GET_CLASS, 2081
ags_fx_fm_synth_channel_get_type, 2082
ags_fx_fm_synth_channel_new, 2081
AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR, 2083
AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS, 2084
AGS_FX_FM_SYNTH_CHANNEL_PROCESSOR_GET_CLASS, 2084
ags_fx_fm_synth_channel_processor_get_type, 2084
ags_fx_fm_synth_channel_processor_new, 2083
AGS_FX_FM_SYNTH_RECYCLING, 2086
AGS_FX_FM_SYNTH_RECYCLING_CLASS, 2086
AGS_FX_FM_SYNTH_RECYCLING_GET_CLASS, 2086
ags_fx_fm_synth_recycling_get_type, 2086
ags_fx_fm_synth_recycling_new, 2085
AGS_FX_HIGH_PASS_AUDIO, 2171
AGS_FX_HIGH_PASS_AUDIO_CLASS, 2171
AGS_FX_HIGH_PASS_AUDIO_GET_CLASS, 2171
ags_fx_high_pass_audio_get_type, 2171
ags_fx_high_pass_audio_new, 2170
AGS_FX_HIGH_PASS_AUDIO_PROCESSOR, 2173
AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS, 2173
AGS_FX_HIGH_PASS_AUDIO_PROCESSOR_GET_CLASS, 2173
ags_fx_high_pass_audio_processor_get_type, 2173
ags_fx_high_pass_audio_processor_new, 2173
AGS_FX_HIGH_PASS_AUDIO_SIGNAL, 2175
AGS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS, 2175
AGS_FX_HIGH_PASS_AUDIO_SIGNAL_GET_CLASS, 2175
ags_fx_high_pass_audio_signal_get_type, 2176
ags_fx_high_pass_audio_signal_new, 2175
AGS_FX_HIGH_PASS_CHANNEL, 2178
AGS_FX_HIGH_PASS_CHANNEL_CLASS, 2178
AGS_FX_HIGH_PASS_CHANNEL_GET_CLASS, 2178
ags_fx_high_pass_channel_get_type, 2178
ags_fx_high_pass_channel_new, 2177
AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR, 2181
AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS, 2181
AGS_FX_HIGH_PASS_CHANNEL_PROCESSOR_GET_CLASS, 2181
ags_fx_high_pass_channel_processor_get_type, 2181
ags_fx_high_pass_channel_processor_new, 2180
AGS_FX_HIGH_PASS_RECYCLING, 2183
AGS_FX_HIGH_PASS_RECYCLING_CLASS, 2183
AGS_FX_HIGH_PASS_RECYCLING_GET_CLASS, 2183
ags_fx_high_pass_recycling_get_type, 2183
ags_fx_high_pass_recycling_new, 2183
AGS_FX_LADSPA_AUDIO, 2219
AGS_FX_LADSPA_AUDIO_CLASS, 2219
AGS_FX_LADSPA_AUDIO_GET_CLASS, 2219
ags_fx_ladspa_audio_get_type, 2220
ags_fx_ladspa_audio_new, 2219
AGS_FX_LADSPA_AUDIO_PROCESSOR, 2222
AGS_FX_LADSPA_AUDIO_PROCESSOR_CLASS, 2222
AGS_FX_LADSPA_AUDIO_PROCESSOR_GET_CLASS, 2222
ags_fx_ladspa_audio_processor_get_type, 2222
ags_fx_ladspa_audio_processor_new, 2221
AGS_FX_LADSPA_AUDIO_SIGNAL, 2224
AGS_FX_LADSPA_AUDIO_SIGNAL_CLASS, 2224
AGS_FX_LADSPA_AUDIO_SIGNAL_GET_CLASS, 2224
ags_fx_ladspa_audio_signal_get_type, 2224
ags_fx_ladspa_audio_signal_new, 2224
AGS_FX_LADSPA_CHANNEL, 2228
AGS_FX_LADSPA_CHANNEL_CLASS, 2228
AGS_FX_LADSPA_CHANNEL_GET_CLASS, 2228
ags_fx_ladspa_channel_get_type, 2228
AGS_FX_LADSPA_CHANNEL_INPUT_DATA, 2226
ags_fx_ladspa_channel_input_data_alloc, 2226
ags_fx_ladspa_channel_input_data_free, 2226
AGS_FX_LADSPA_CHANNEL_INPUT_DATA_GET_STRCT_M 2226
ags_fx_ladspa_channel_load_plugin, 2227
ags_fx_ladspa_channel_load_port, 2227
ags_fx_ladspa_channel_new, 2227
AGS_FX_LADSPA_CHANNEL_PROCESSOR, 2230
AGS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS, 2230
AGS_FX_LADSPA_CHANNEL_PROCESSOR_GET_CLASS, 2230
ags_fx_ladspa_channel_processor_get_type, 2231
ags_fx_ladspa_channel_processor_new, 2230
AGS_FX_LADSPA_RECYCLING, 2232
AGS_FX_LADSPA_RECYCLING_CLASS, 2232
AGS_FX_LADSPA_RECYCLING_GET_CLASS, 2233
ags_fx_ladspa_recycling_get_type, 2233
ags_fx_ladspa_recycling_new, 2232
AGS_FX_LFO_AUDIO, 2235
AGS_FX_LFO_AUDIO_CLASS, 2235
AGS_FX_LFO_AUDIO_GET_CLASS, 2235
ags_fx_lfo_audio_get_type, 2235
ags_fx_lfo_audio_new, 2234
AGS_FX_LFO_AUDIO_PROCESSOR, 2237
AGS_FX_LFO_AUDIO_PROCESSOR_CLASS, 2237

AGS_FX_LFO_AUDIO_PROCESSOR_GET_CLASS, 2237
ags_fx_lfo_audio_processor_get_type, 2237
ags_fx_lfo_audio_processor_new, 2237
AGS_FX_LFO_AUDIO_SIGNAL, 2239
AGS_FX_LFO_AUDIO_SIGNAL_CLASS, 2239
AGS_FX_LFO_AUDIO_SIGNAL_GET_CLASS, 2239
ags_fx_lfo_audio_signal_get_type, 2240
ags_fx_lfo_audio_signal_new, 2239
AGS_FX_LFO_CHANNEL, 2242
AGS_FX_LFO_CHANNEL_CLASS, 2243
AGS_FX_LFO_CHANNEL_GET_CLASS, 2243
ags_fx_lfo_channel_get_type, 2243
ags_fx_lfo_channel_input_data_alloc, 2242
ags_fx_lfo_channel_input_data_free, 2242
ags_fx_lfo_channel_new, 2242
AGS_FX_LFO_CHANNEL_PROCESSOR, 2246
AGS_FX_LFO_CHANNEL_PROCESSOR_CLASS, 2246
AGS_FX_LFO_CHANNEL_PROCESSOR_GET_CLASS, 2246
ags_fx_lfo_channel_processor_get_type, 2247
ags_fx_lfo_channel_processor_new, 2246
AGS_FX_LFO_RECYCLING, 2249
AGS_FX_LFO_RECYCLING_CLASS, 2249
AGS_FX_LFO_RECYCLING_GET_CLASS, 2249
ags_fx_lfo_recycling_get_type, 2249
ags_fx_lfo_recycling_new, 2248
AGS_FX_LOW_PASS_AUDIO, 2156
AGS_FX_LOW_PASS_AUDIO_CLASS, 2156
AGS_FX_LOW_PASS_AUDIO_GET_CLASS, 2156
ags_fx_low_pass_audio_get_type, 2156
ags_fx_low_pass_audio_new, 2156
AGS_FX_LOW_PASS_AUDIO_PROCESSOR, 2158
AGS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS, 2158
AGS_FX_LOW_PASS_AUDIO_PROCESSOR_GET_CLASS, 2158
ags_fx_low_pass_audio_processor_get_type, 2159
ags_fx_low_pass_audio_processor_new, 2158
AGS_FX_LOW_PASS_AUDIO_SIGNAL, 2161
AGS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS, 2161
AGS_FX_LOW_PASS_AUDIO_SIGNAL_GET_CLASS, 2161
ags_fx_low_pass_audio_signal_get_type, 2161
ags_fx_low_pass_audio_signal_new, 2160
AGS_FX_LOW_PASS_CHANNEL, 2163
AGS_FX_LOW_PASS_CHANNEL_CLASS, 2163
AGS_FX_LOW_PASS_CHANNEL_GET_CLASS, 2163
ags_fx_low_pass_channel_get_type, 2164
ags_fx_low_pass_channel_new, 2163
AGS_FX_LOW_PASS_CHANNEL_PROCESSOR, 2166
AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS, 2166
AGS_FX_LOW_PASS_CHANNEL_PROCESSOR_GET_CLASS, 2166
ags_fx_low_pass_channel_processor_get_type, 2167
ags_fx_low_pass_channel_processor_new, 2166
AGS_FX_LOW_PASS_RECYCLING, 2168
AGS_FX_LOW_PASS_RECYCLING_CLASS, 2168
AGS_FX_LOW_PASS_RECYCLING_GET_CLASS, 2169
ags_fx_low_pass_recycling_get_type, 2169
ags_fx_low_pass_recycling_new, 2168
AGS_FX_LV2_AUDIO, 2017
ags_fx_lv2_audio_change_program, 2016
AGS_FX_LV2_AUDIO_CHANNEL_DATA, 2013
ags_fx_lv2_audio_channel_data_alloc, 2014
ags_fx_lv2_audio_channel_data_free, 2014
AGS_FX_LV2_AUDIO_CHANNEL_DATA_GET_STRCT_MUT 2013
AGS_FX_LV2_AUDIO_CLASS, 2017
AGS_FX_LV2_AUDIO_DEFAULT_MIDI_LENGHT, 2018
AGS_FX_LV2_AUDIO_GET_CLASS, 2017
ags_fx_lv2_audio_get_type, 2017
AGS_FX_LV2_AUDIO_INPUT_DATA, 2013
ags_fx_lv2_audio_input_data_alloc, 2014
ags_fx_lv2_audio_input_data_free, 2015
AGS_FX_LV2_AUDIO_INPUT_DATA_GET_STRCT_MUTEX, 2013
ags_fx_lv2_audio_load_plugin, 2016
ags_fx_lv2_audio_load_port, 2016
ags_fx_lv2_audio_new, 2017
AGS_FX_LV2_AUDIO_PROCESSOR, 2020
AGS_FX_LV2_AUDIO_PROCESSOR_CLASS, 2020
AGS_FX_LV2_AUDIO_PROCESSOR_GET_CLASS, 2021
ags_fx_lv2_audio_processor_get_type, 2021
ags_fx_lv2_audio_processor_new, 2020
AGS_FX_LV2_AUDIO_SCOPE_DATA, 2013
ags_fx_lv2_audio_scope_data_alloc, 2013
ags_fx_lv2_audio_scope_data_free, 2014
AGS_FX_LV2_AUDIO_SCOPE_DATA_GET_STRCT_MUTEX, 2013
ags_fx_lv2_audio_set_flags, 2015
AGS_FX_LV2_AUDIO_SIGNAL, 2023
AGS_FX_LV2_AUDIO_SIGNAL_CLASS, 2023
AGS_FX_LV2_AUDIO_SIGNAL_GET_CLASS, 2023
ags_fx_lv2_audio_signal_get_type, 2023
ags_fx_lv2_audio_signal_new, 2022
ags_fx_lv2_audio_test_flags, 2015
ags_fx_lv2_audio_unset_flags, 2016
AGS_FX_LV2_CHANNEL, 2026
AGS_FX_LV2_CHANNEL_CLASS, 2026
AGS_FX_LV2_CHANNEL_GET_CLASS, 2026
ags_fx_lv2_channel_get_type, 2027
AGS_FX_LV2_CHANNEL_INPUT_DATA, 2025
ags_fx_lv2_channel_input_data_alloc, 2025
ags_fx_lv2_channel_input_data_free, 2025
AGS_FX_LV2_CHANNEL_INPUT_DATA_GET_STRCT_MUTI 2025
ags_fx_lv2_channel_load_plugin, 2025
ags_fx_lv2_channel_load_port, 2026
ags_fx_lv2_channel_new, 2026
AGS_FX_LV2_CHANNEL_PROCESSOR, 2029
AGS_FX_LV2_CHANNEL_PROCESSOR_CLASS, 2029
AGS_FX_LV2_CHANNEL_PROCESSOR_GET_CLASS, 2029
ags_fx_lv2_channel_processor_get_type, 2029
ags_fx_lv2_channel_processor_new, 2028
AGS_FX_LV2_RECYCLING, 2031

AGS_FX_LV2_RECYCLING_CLASS, 2031
AGS_FX_LV2_RECYCLING_GET_CLASS, 2031
ags_fx_lv2_recycling_get_type, 2032
ags_fx_lv2_recycling_new, 2031
AGS_FX_NOTATION_AUDIO, 1974
ags_fx_notation_audio_add_feed_note, 1973
AGS_FX_NOTATION_AUDIO_CLASS, 1974
AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_END, 1975
AGS_FX_NOTATION_AUDIO_DEFAULT_LOOP_START, 1975
AGS_FX_NOTATION_AUDIO_GET_CLASS, 1974
ags_fx_notation_audio_get_feed_note, 1973
ags_fx_notation_audio_get_pattern_mode, 1972
ags_fx_notation_audio_get_type, 1975
ags_fx_notation_audio_new, 1974
AGS_FX_NOTATION_AUDIO_PROCESSOR, 1980
AGS_FX_NOTATION_AUDIO_PROCESSOR_CLASS, 1980
ags_fx_notation_audio_processor_counter_change, 1979
AGS_FX_NOTATION_AUDIO_PROCESSOR_DEFAULT_KEY_ON_VELOCITY, 1980
ags_fx_notation_audio_processor_feed, 1979
AGS_FX_NOTATION_AUDIO_PROCESSOR_GET_CLASS, 1980
ags_fx_notation_audio_processor_get_type, 1980
ags_fx_notation_audio_processor_key_off, 1979
ags_fx_notation_audio_processor_key_on, 1978
ags_fx_notation_audio_processor_key_pressure, 1979
ags_fx_notation_audio_processor_new, 1979
ags_fx_notation_audio_processor_play, 1979
ags_fx_notation_audio_processor_record, 1979
ags_fx_notation_audio_remove_feed_note, 1974
ags_fx_notation_audio_set_flags, 1972
ags_fx_notation_audio_set_pattern_mode, 1973
AGS_FX_NOTATION_AUDIO_SIGNAL, 1983
AGS_FX_NOTATION_AUDIO_SIGNAL_CLASS, 1983
AGS_FX_NOTATION_AUDIO_SIGNAL_GET_CLASS, 1984
ags_fx_notation_audio_signal_get_type, 1984
ags_fx_notation_audio_signal_new, 1983
ags_fx_notation_audio_signal_notify_remove, 1983
ags_fx_notation_audio_signal_stream_feed, 1982
ags_fx_notation_audio_test_flags, 1971
ags_fx_notation_audio_unset_flags, 1972
AGS_FX_NOTATION_CHANNEL, 1986
AGS_FX_NOTATION_CHANNEL_CLASS, 1986
AGS_FX_NOTATION_CHANNEL_GET_CLASS, 1986
ags_fx_notation_channel_get_type, 1986
ags_fx_notation_channel_new, 1986
AGS_FX_NOTATION_CHANNEL_PROCESSOR, 1988
AGS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS, 1988
AGS_FX_NOTATION_CHANNEL_PROCESSOR_GET_CLASS, 1989
ags_fx_notation_channel_processor_get_type, 1989
ags_fx_notation_channel_processor_new, 1988
AGS_FX_NOTATION_RECYCLING, 1991
AGS_FX_NOTATION_RECYCLING_CLASS, 1991
AGS_FX_NOTATION_RECYCLING_GET_CLASS, 1991
ags_fx_notation_recycling_get_type, 1991
ags_fx_notation_recycling_new, 1990
AGS_FX_PATTERN_AUDIO, 1953
AGS_FX_PATTERN_AUDIO_CLASS, 1954
AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_END, 1954
AGS_FX_PATTERN_AUDIO_DEFAULT_LOOP_START, 1954
AGS_FX_PATTERN_AUDIO_GET_CLASS, 1954
ags_fx_pattern_audio_get_type, 1954
ags_fx_pattern_audio_new, 1953
AGS_FX_PATTERN_AUDIO_PROCESSOR, 1958
AGS_FX_PATTERN_AUDIO_PROCESSOR_CLASS, 1958
ags_fx_pattern_audio_processor_counter_change, 1958
AGS_FX_PATTERN_AUDIO_PROCESSOR_DEFAULT_KEY_C 1959
AGS_FX_PATTERN_AUDIO_PROCESSOR_GET_CLASS, 1959
ags_fx_pattern_audio_processor_get_type, 1959
ags_fx_pattern_audio_processor_key_on, 1958
ags_fx_pattern_audio_processor_new, 1958
ags_fx_pattern_audio_processor_play, 1958
AGS_FX_PATTERN_AUDIO_SIGNAL, 1961
AGS_FX_PATTERN_AUDIO_SIGNAL_CLASS, 1962
AGS_FX_PATTERN_AUDIO_SIGNAL_GET_CLASS, 1962
ags_fx_pattern_audio_signal_get_type, 1962
ags_fx_pattern_audio_signal_new, 1961
ags_fx_pattern_audio_signal_notify_remove, 1961
ags_fx_pattern_audio_signal_stream_feed, 1961
AGS_FX_PATTERN_CHANNEL, 1964
AGS_FX_PATTERN_CHANNEL_CLASS, 1964
AGS_FX_PATTERN_CHANNEL_GET_CLASS, 1964
ags_fx_pattern_channel_get_type, 1965
ags_fx_pattern_channel_new, 1964
AGS_FX_PATTERN_CHANNEL_PROCESSOR, 1966
AGS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS, 1966
AGS_FX_PATTERN_CHANNEL_PROCESSOR_GET_CLASS, 1967
ags_fx_pattern_channel_processor_get_type, 1967
ags_fx_pattern_channel_processor_new, 1966
AGS_FX_PATTERN_RECYCLING, 1969
AGS_FX_PATTERN_RECYCLING_CLASS, 1969
AGS_FX_PATTERN_RECYCLING_GET_CLASS, 1969
ags_fx_pattern_recycling_get_type, 1969
ags_fx_pattern_recycling_new, 1968
AGS_FX_PEAK_AUDIO, 2108
AGS_FX_PEAK_AUDIO_CLASS, 2108
AGS_FX_PEAK_AUDIO_GET_CLASS, 2108
ags_fx_peak_audio_get_type, 2109
ags_fx_peak_audio_new, 2108
AGS_FX_PEAK_AUDIO_PROCESSOR, 2111
AGS_FX_PEAK_AUDIO_PROCESSOR_CLASS, 2111
AGS_FX_PEAK_AUDIO_PROCESSOR_GET_CLASS, 2111
ags_fx_peak_audio_processor_get_type, 2111
ags_fx_peak_audio_processor_new, 2110
AGS_FX_PEAK_AUDIO_SIGNAL, 2113
AGS_FX_PEAK_AUDIO_SIGNAL_CLASS, 2113

AGS_FX_PEAK_AUDIO_SIGNAL_GET_CLASS, 2113
ags_fx_peak_audio_signal_get_type, 2113
ags_fx_peak_audio_signal_new, 2113
AGS_FX_PEAK_CHANNEL, 2116
AGS_FX_PEAK_CHANNEL_CLASS, 2116
AGS_FX_PEAK_CHANNEL_GET_CLASS, 2116
ags_fx_peak_channel_get_type, 2117
AGS_FX_PEAK_CHANNEL_INPUT_DATA, 2115
ags_fx_peak_channel_input_data_alloc, 2115
ags_fx_peak_channel_input_data_free, 2116
AGS_FX_PEAK_CHANNEL_INPUT_DATA_GET_STRCT_MUTEX, 2115
ags_fx_peak_channel_new, 2116
AGS_FX_PEAK_CHANNEL_PROCESSOR, 2119
AGS_FX_PEAK_CHANNEL_PROCESSOR_CLASS, 2119
AGS_FX_PEAK_CHANNEL_PROCESSOR_GET_CLASS, 2119
ags_fx_peak_channel_processor_get_type, 2119
ags_fx_peak_channel_processor_new, 2118
AGS_FX_PEAK_RECYCLING, 2121
AGS_FX_PEAK_RECYCLING_CLASS, 2121
AGS_FX_PEAK_RECYCLING_GET_CLASS, 2121
ags_fx_peak_recycling_get_type, 2122
ags_fx_peak_recycling_new, 2121
AGS_FX_PLAYBACK_AUDIO, 1916
ags_fx_playback_audio_add_feed_audio_signal, 1914
ags_fx_playback_audio_add_master_audio_signal, 1915
AGS_FX_PLAYBACK_AUDIO_CLASS, 1916
AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_END, 1917
AGS_FX_PLAYBACK_AUDIO_DEFAULT_LOOP_START, 1917
AGS_FX_PLAYBACK_AUDIO_GET_CLASS, 1916
ags_fx_playback_audio_get_feed_audio_signal, 1913
ags_fx_playback_audio_get_master_audio_signal, 1914
ags_fx_playback_audio_get_type, 1917
ags_fx_playback_audio_new, 1916
ags_fx_playback_audio_open_audio_file, 1915
AGS_FX_PLAYBACK_AUDIO_PROCESSOR, 1924
AGS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS, 1924
ags_fx_playback_audio_processor_counter_change, 1923
ags_fx_playback_audio_processor_data_get, 1922
ags_fx_playback_audio_processor_data_put, 1922
ags_fx_playback_audio_processor_feed, 1923
AGS_FX_PLAYBACK_AUDIO_PROCESSOR_GET_CLASS, 1924
ags_fx_playback_audio_processor_get_type, 1924
ags_fx_playback_audio_processor_master, 1923
ags_fx_playback_audio_processor_new, 1923
ags_fx_playback_audio_processor_play, 1923
ags_fx_playback_audio_processor_record, 1923
ags_fx_playback_audio_remove_feed_audio_signal, 1914
ags_fx_playback_audio_remove_master_audio_signal, 1915
ags_fx_playback_audio_set_flags, 1913
AGS_FX_PLAYBACK_AUDIO_SIGNAL, 1927
AGS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS, 1927
AGS_FX_PLAYBACK_AUDIO_SIGNAL_GET_CLASS, 1927
ags_fx_playback_audio_signal_get_type, 1927
ags_fx_playback_audio_signal_new, 1926
ags_fx_playback_audio_test_flags, 1912
ags_fx_playback_audio_unset_flags, 1913
AGS_FX_PLAYBACK_CHANNEL, 1929
AGS_FX_PLAYBACK_CHANNEL_CLASS, 1929
AGS_FX_PLAYBACK_CHANNEL_GET_CLASS, 1929
ags_fx_playback_channel_get_type, 1929
ags_fx_playback_channel_new, 1929
AGS_FX_PLAYBACK_CHANNEL_PROCESSOR, 1931
AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS, 1931
AGS_FX_PLAYBACK_CHANNEL_PROCESSOR_GET_CLASS, 1931
ags_fx_playback_channel_processor_get_type, 1932
ags_fx_playback_channel_processor_new, 1931
AGS_FX_PLAYBACK_RECYCLING, 1934
AGS_FX_PLAYBACK_RECYCLING_CLASS, 1934
AGS_FX_PLAYBACK_RECYCLING_GET_CLASS, 1934
ags_fx_playback_recycling_get_type, 1934
ags_fx_playback_recycling_new, 1933
AGS_FX_SF2_SYNTH_AUDIO, 2090
ags_fx_sf2_synth_audio_channel_data_alloc, 2089
ags_fx_sf2_synth_audio_channel_data_free, 2089
AGS_FX_SF2_SYNTH_AUDIO_CLASS, 2090
AGS_FX_SF2_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE, 2091
AGS_FX_SF2_SYNTH_AUDIO_GET_CLASS, 2090
ags_fx_sf2_synth_audio_get_type, 2091
ags_fx_sf2_synth_audio_input_data_alloc, 2089
ags_fx_sf2_synth_audio_input_data_free, 2090
ags_fx_sf2_synth_audio_new, 2090
AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR, 2096
AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS, 2096
AGS_FX_SF2_SYNTH_AUDIO_PROCESSOR_GET_CLASS, 2096
ags_fx_sf2_synth_audio_processor_get_type, 2097
ags_fx_sf2_synth_audio_processor_new, 2096
ags_fx_sf2_synth_audio_scope_data_alloc, 2088
ags_fx_sf2_synth_audio_scope_data_free, 2088
AGS_FX_SF2_SYNTH_AUDIO_SIGNAL, 2098
AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS, 2098
AGS_FX_SF2_SYNTH_AUDIO_SIGNAL_GET_CLASS, 2099
ags_fx_sf2_synth_audio_signal_get_type, 2099
ags_fx_sf2_synth_audio_signal_new, 2098
AGS_FX_SF2_SYNTH_CHANNEL, 2101
AGS_FX_SF2_SYNTH_CHANNEL_CLASS, 2101
AGS_FX_SF2_SYNTH_CHANNEL_GET_CLASS, 2101
ags_fx_sf2_synth_channel_get_type, 2101
ags_fx_sf2_synth_channel_new, 2100
AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR, 2103
AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS, 2103
AGS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_GET_CLAS 2103
ags_fx_sf2_synth_channel_processor_get_type, 2103
ags_fx_sf2_synth_channel_processor_new, 2103

AGS_FX_SF2_SYNTH_RECYCLING, 2105
AGS_FX_SF2_SYNTH_RECYCLING_CLASS, 2105
AGS_FX_SF2_SYNTH_RECYCLING_GET_CLASS, 2105
ags_fx_sf2_synth_recycling_get_type, 2106
ags_fx_sf2_synth_recycling_new, 2105
AGS_FX_SYNTH_AUDIO, 2036
ags_fx_synth_audio_channel_data_alloc, 2035
ags_fx_synth_audio_channel_data_free, 2035
AGS_FX_SYNTH_AUDIO_CLASS, 2036
AGS_FX_SYNTH_AUDIO_DEFAULT_BUFFER_SIZE, 2037
AGS_FX_SYNTH_AUDIO_GET_CLASS, 2036
ags_fx_synth_audio_get_type, 2037
ags_fx_synth_audio_input_data_alloc, 2035
ags_fx_synth_audio_input_data_free, 2036
ags_fx_synth_audio_new, 2036
AGS_FX_SYNTH_AUDIO_PROCESSOR, 2049
AGS_FX_SYNTH_AUDIO_PROCESSOR_CLASS, 2050
AGS_FX_SYNTH_AUDIO_PROCESSOR_GET_CLASS, 2050
ags_fx_synth_audio_processor_get_type, 2050
ags_fx_synth_audio_processor_new, 2049
ags_fx_synth_audio_scope_data_alloc, 2034
ags_fx_synth_audio_scope_data_free, 2035
AGS_FX_SYNTH_AUDIO_SIGNAL, 2052
AGS_FX_SYNTH_AUDIO_SIGNAL_CLASS, 2052
AGS_FX_SYNTH_AUDIO_SIGNAL_GET_CLASS, 2052
ags_fx_synth_audio_signal_get_type, 2052
ags_fx_synth_audio_signal_new, 2051
AGS_FX_SYNTH_CHANNEL, 2054
AGS_FX_SYNTH_CHANNEL_CLASS, 2054
AGS_FX_SYNTH_CHANNEL_GET_CLASS, 2054
ags_fx_synth_channel_get_type, 2055
ags_fx_synth_channel_new, 2054
AGS_FX_SYNTH_CHANNEL_PROCESSOR, 2056
AGS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS, 2057
AGS_FX_SYNTH_CHANNEL_PROCESSOR_GET_CLASS, 2057
ags_fx_synth_channel_processor_get_type, 2057
ags_fx_synth_channel_processor_new, 2056
AGS_FX_SYNTH_RECYCLING, 2059
AGS_FX_SYNTH_RECYCLING_CLASS, 2059
AGS_FX_SYNTH_RECYCLING_GET_CLASS, 2059
ags_fx_synth_recycling_get_type, 2059
ags_fx_synth_recycling_new, 2058
AGS_FX_TWO_PASS_ALIASE_AUDIO, 2251
AGS_FX_TWO_PASS_ALIASE_AUDIO_CLASS, 2251
AGS_FX_TWO_PASS_ALIASE_AUDIO_GET_CLASS, 2251
ags_fx_two_pass_aliase_audio_get_type, 2251
ags_fx_two_pass_aliase_audio_new, 2251
AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR, 2253
AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS, 2253
AGS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_GET_CLASS, 2253
ags_fx_two_pass_aliase_audio_processor_get_type, 2254
ags_fx_two_pass_aliase_audio_processor_new, 2253
AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL, 2256
AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS, 2256
AGS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_GET_CLAS 2256
ags_fx_two_pass_aliase_audio_signal_get_type, 2256
ags_fx_two_pass_aliase_audio_signal_new, 2255
AGS_FX_TWO_PASS_ALIASE_CHANNEL, 2262
AGS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS, 2262
AGS_FX_TWO_PASS_ALIASE_CHANNEL_GET_CLASS, 2262
ags_fx_two_pass_aliase_channel_get_type, 2263
AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA, 2258
ags_fx_two_pass_aliase_channel_input_data_alloc, 2258
ags_fx_two_pass_aliase_channel_input_data_free, 2259
AGS_FX_TWO_PASS_ALIASE_CHANNEL_INPUT_DATA_GE 2258
ags_fx_two_pass_aliase_channel_input_data_get_strct_mutex, 2259
ags_fx_two_pass_aliase_channel_input_get_a_buffer, 2260
ags_fx_two_pass_aliase_channel_input_get_a_mix, 2261
ags_fx_two_pass_aliase_channel_input_get_b_buffer, 2260
ags_fx_two_pass_aliase_channel_input_get_b_mix, 2261
ags_fx_two_pass_aliase_channel_input_get_final_mix, 2262
ags_fx_two_pass_aliase_channel_input_get_orig_buffer, 2260
ags_fx_two_pass_aliase_channel_input_get_parent, 2259
ags_fx_two_pass_aliase_channel_new, 2262
AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR, 2266
AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CL 2266
AGS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_GE 2266
ags_fx_two_pass_aliase_channel_processor_get_type, 2266
ags_fx_two_pass_aliase_channel_processor_new, 2265
AGS_FX_TWO_PASS_ALIASE_RECYCLING, 2268
AGS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS, 2268
AGS_FX_TWO_PASS_ALIASE_RECYCLING_GET_CLASS, 2268
ags_fx_two_pass_aliase_recycling_get_type, 2269
ags_fx_two_pass_aliase_recycling_new, 2268
AGS_FX_VOLUME_AUDIO, 2141
AGS_FX_VOLUME_AUDIO_CLASS, 2141
AGS_FX_VOLUME_AUDIO_GET_CLASS, 2142
ags_fx_volume_audio_get_type, 2142
ags_fx_volume_audio_new, 2141
AGS_FX_VOLUME_AUDIO_PROCESSOR, 2144
AGS_FX_VOLUME_AUDIO_PROCESSOR_CLASS, 2144
AGS_FX_VOLUME_AUDIO_PROCESSOR_GET_CLASS, 2144
ags_fx_volume_audio_processor_get_type, 2144
ags_fx_volume_audio_processor_new, 2143
AGS_FX_VOLUME_AUDIO_SIGNAL, 2146
AGS_FX_VOLUME_AUDIO_SIGNAL_CLASS, 2146
AGS_FX_VOLUME_AUDIO_SIGNAL_GET_CLASS, 2146
ags_fx_volume_audio_signal_get_type, 2147
ags_fx_volume_audio_signal_new, 2146

AGS_FX_VOLUME_CHANNEL, 2148
AGS_FX_VOLUME_CHANNEL_CLASS, 2149
AGS_FX_VOLUME_CHANNEL_GET_CLASS, 2149
ags_fx_volume_channel_get_type, 2149
ags_fx_volume_channel_new, 2148
AGS_FX_VOLUME_CHANNEL_PROCESSOR, 2151
AGS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS, 2151
AGS_FX_VOLUME_CHANNEL_PROCESSOR_GET_CLASS, 2152
ags_fx_volume_channel_processor_get_type, 2152
ags_fx_volume_channel_processor_new, 2151
AGS_FX_VOLUME_RECYCLING, 2154
AGS_FX_VOLUME_RECYCLING_CLASS, 2154
AGS_FX_VOLUME_RECYCLING_GET_CLASS, 2154
ags_fx_volume_recycling_get_type, 2154
ags_fx_volume_recycling_new, 2153
AgsFxAnalyseAudio, 2124
AgsFxAnalyseAudioClass, 2124
AgsFxAnalyseAudioProcessor, 2126
AgsFxAnalyseAudioProcessorClass, 2127
AgsFxAnalyseAudioSignal, 2129
AgsFxAnalyseAudioSignalClass, 2129
AgsFxAnalyseChannel, 2134
AgsFxAnalyseChannel:frequency, 2134
AgsFxAnalyseChannel:magnitude, 2135
AgsFxAnalyseChannelClass, 2134
AgsFxAnalyseChannelInputData, 2134
AgsFxAnalyseChannelProcessor, 2137
AgsFxAnalyseChannelProcessorClass, 2137
AgsFxAnalyseRecycling, 2139
AgsFxAnalyseRecyclingClass, 2139
AgsFxBufferAudio, 1937
AgsFxBufferAudioClass, 1937
AgsFxBufferAudioProcessor, 1939
AgsFxBufferAudioProcessorClass, 1939
AgsFxBufferAudioSignal, 1941
AgsFxBufferAudioSignalClass, 1942
AgsFxBufferChannel, 1946
AgsFxBufferChannelClass, 1947
AgsFxBufferChannelInputData, 1946
AgsFxBufferChannelProcessor, 1949
AgsFxBufferChannelProcessorClass, 1949
AgsFxBufferRecycling, 1951
AgsFxBufferRecyclingClass, 1951
AgsFxDssiAudio, 1999
AgsFxDssiAudioChannelData, 1999
AgsFxDssiAudioClass, 1999
AgsFxDssiAudioFlags, 1998
AgsFxDssiAudioInputData, 1999
AgsFxDssiAudioProcessor, 2002
AgsFxDssiAudioProcessorClass, 2002
AgsFxDssiAudioScopeData, 1998
AgsFxDssiAudioSignal, 2004
AgsFxDssiAudioSignalClass, 2004
AgsFxDssiChannel, 2007
AgsFxDssiChannelClass, 2007
AgsFxDssiChannelProcessor, 2009
AgsFxDssiChannelProcessorClass, 2009
AgsFxDssiRecycling, 2011
AgsFxDssiRecyclingClass, 2011
AgsFxEnvelopeAudio, 2204
AgsFxEnvelopeAudioClass, 2204
AgsFxEnvelopeAudioProcessor, 2206
AgsFxEnvelopeAudioProcessorClass, 2206
AgsFxEnvelopeAudioSignal, 2208
AgsFxEnvelopeAudioSignalClass, 2209
AgsFxEnvelopeChannel, 2211
AgsFxEnvelopeChannel:attack, 2211
AgsFxEnvelopeChannel:decay, 2212
AgsFxEnvelopeChannel:fixed-length, 2212
AgsFxEnvelopeChannel:ratio, 2212
AgsFxEnvelopeChannel:release, 2212
AgsFxEnvelopeChannel:sustain, 2212
AgsFxEnvelopeChannelClass, 2211
AgsFxEnvelopeChannelMode, 2211
AgsFxEnvelopeChannelProcessor, 2215
AgsFxEnvelopeChannelProcessorClass, 2215
AgsFxEnvelopeRecycling, 2217
AgsFxEnvelopeRecyclingClass, 2217
AgsFxEq10Audio, 2186
AgsFxEq10AudioClass, 2186
AgsFxEq10AudioProcessor, 2188
AgsFxEq10AudioProcessorClass, 2189
AgsFxEq10AudioSignal, 2191
AgsFxEq10AudioSignalClass, 2191
AgsFxEq10Channel, 2195
AgsFxEq10Channel:peak-112hz, 2195
AgsFxEq10Channel:peak-14336hz, 2195
AgsFxEq10Channel:peak-1792hz, 2195
AgsFxEq10Channel:peak-224hz, 2195
AgsFxEq10Channel:peak-28hz, 2196
AgsFxEq10Channel:peak-3584hz, 2196
AgsFxEq10Channel:peak-448hz, 2196
AgsFxEq10Channel:peak-56hz, 2196
AgsFxEq10Channel:peak-7168hz, 2196
AgsFxEq10Channel:peak-896hz, 2197
AgsFxEq10Channel:pressure, 2197
AgsFxEq10ChannelClass, 2195
AgsFxEq10ChannelInputData, 2194
AgsFxEq10ChannelProcessor, 2199
AgsFxEq10ChannelProcessorClass, 2199
AgsFxEq10Recycling, 2201
AgsFxEq10RecyclingClass, 2202
AgsFxFactory, 1321
AgsFxFactoryClass, 1321
AgsFxFactoryCreateFlags, 1320
AgsFxFMSynthAudio, 2066
AgsFxFMSynthAudio:chorus-delay, 2066
AgsFxFMSynthAudio:chorus-depth, 2066
AgsFxFMSynthAudio:chorus-enabled, 2066
AgsFxFMSynthAudio:chorus-input-volume, 2066
AgsFxFMSynthAudio:chorus-lfo-frequency, 2067
AgsFxFMSynthAudio:chorus-lfo-oscillator, 2067
AgsFxFMSynthAudio:chorus-mix, 2067

AgsFxFMSynthAudio:chorus-output-volume, 2067
AgsFxFMSynthAudio:chorus-pitch-type, 2067
AgsFxFMSynthAudio:high-pass-enabled, 2068
AgsFxFMSynthAudio:high-pass-filter-gain, 2068
AgsFxFMSynthAudio:high-pass-q-lin, 2068
AgsFxFMSynthAudio:low-pass-enabled, 2068
AgsFxFMSynthAudio:low-pass-filter-gain, 2068
AgsFxFMSynthAudio:low-pass-q-lin, 2069
AgsFxFMSynthAudio:noise-gain, 2069
AgsFxFMSynthAudio:pitch-tuning, 2069
AgsFxFMSynthAudio:pitch-type, 2069
AgsFxFMSynthAudio:sequencer-enabled, 2069
AgsFxFMSynthAudio:sequencer-sign, 2070
AgsFxFMSynthAudio:synth-0-key, 2070
AgsFxFMSynthAudio:synth-0-lfo-depth, 2070
AgsFxFMSynthAudio:synth-0-lfo-frequency, 2070
AgsFxFMSynthAudio:synth-0-lfo-oscillator, 2070
AgsFxFMSynthAudio:synth-0-lfo-tuning, 2071
AgsFxFMSynthAudio:synth-0-octave, 2071
AgsFxFMSynthAudio:synth-0-oscillator, 2071
AgsFxFMSynthAudio:synth-0-phase, 2071
AgsFxFMSynthAudio:synth-0-volume, 2071
AgsFxFMSynthAudio:synth-1-key, 2072
AgsFxFMSynthAudio:synth-1-lfo-depth, 2072
AgsFxFMSynthAudio:synth-1-lfo-frequency, 2072
AgsFxFMSynthAudio:synth-1-lfo-oscillator, 2072
AgsFxFMSynthAudio:synth-1-lfo-tuning, 2072
AgsFxFMSynthAudio:synth-1-octave, 2073
AgsFxFMSynthAudio:synth-1-oscillator, 2073
AgsFxFMSynthAudio:synth-1-phase, 2073
AgsFxFMSynthAudio:synth-1-volume, 2073
AgsFxFMSynthAudio:synth-2-key, 2073
AgsFxFMSynthAudio:synth-2-lfo-depth, 2074
AgsFxFMSynthAudio:synth-2-lfo-frequency, 2074
AgsFxFMSynthAudio:synth-2-lfo-oscillator, 2074
AgsFxFMSynthAudio:synth-2-lfo-tuning, 2074
AgsFxFMSynthAudio:synth-2-octave, 2074
AgsFxFMSynthAudio:synth-2-oscillator, 2075
AgsFxFMSynthAudio:synth-2-phase, 2075
AgsFxFMSynthAudio:synth-2-volume, 2075
AgsFxFMSynthAudioChannelData, 2065
AgsFxFMSynthAudioClass, 2066
AgsFxFMSynthAudioInputData, 2065
AgsFxFMSynthAudioProcessor, 2077
AgsFxFMSynthAudioProcessorClass, 2077
AgsFxFMSynthAudioScopeData, 2065
AgsFxFMSynthAudioSignal, 2080
AgsFxFMSynthAudioSignalClass, 2080
AgsFxFMSynthChannel, 2082
AgsFxFMSynthChannelClass, 2082
AgsFxFMSynthChannelProcessor, 2084
AgsFxFMSynthChannelProcessorClass, 2084
AgsFxFMSynthRecycling, 2087
AgsFxFMSynthRecyclingClass, 2087
AgsFxHighPassAudio, 2171
AgsFxHighPassAudioClass, 2172
AgsFxHighPassAudioProcessor, 2174
AgsFxHighPassAudioProcessorClass, 2174
AgsFxHighPassAudioSignal, 2176
AgsFxHighPassAudioSignalClass, 2176
AgsFxHighPassChannel, 2179
AgsFxHighPassChannel:enabled, 2179
AgsFxHighPassChannel:filter-gain, 2179
AgsFxHighPassChannel:q-lin, 2179
AgsFxHighPassChannelClass, 2179
AgsFxHighPassChannelProcessor, 2181
AgsFxHighPassChannelProcessorClass, 2182
AgsFxHighPassRecycling, 2184
AgsFxHighPassRecyclingClass, 2184
AgsFxLadspaAudio, 2220
AgsFxLadspaAudioClass, 2220
AgsFxLadspaAudioProcessor, 2222
AgsFxLadspaAudioProcessorClass, 2223
AgsFxLadspaAudioSignal, 2225
AgsFxLadspaAudioSignalClass, 2225
AgsFxLadspaChannel, 2229
AgsFxLadspaChannelClass, 2229
AgsFxLadspaChannelInputData, 2228
AgsFxLadspaChannelProcessor, 2231
AgsFxLadspaChannelProcessorClass, 2231
AgsFxLadspaRecycling, 2233
AgsFxLadspaRecyclingClass, 2233
AgsFxLfoAudio, 2235
AgsFxLfoAudioClass, 2236
AgsFxLfoAudioProcessor, 2238
AgsFxLfoAudioProcessorClass, 2238
AgsFxLfoAudioSignal, 2240
AgsFxLfoAudioSignalClass, 2240
AgsFxLfoChannel, 2243
AgsFxLfoChannel:enabled, 2244
AgsFxLfoChannel:lfo-depth, 2244
AgsFxLfoChannel:lfo-freq, 2244
AgsFxLfoChannel:lfo-phase, 2244
AgsFxLfoChannel:lfo-tuning, 2245
AgsFxLfoChannel:lfo-wave, 2245
AgsFxLfoChannelClass, 2244
AgsFxLfoChannelInputData, 2243
AgsFxLfoChannelProcessor, 2247
AgsFxLfoChannelProcessorClass, 2247
AgsFxLfoRecycling, 2249
AgsFxLfoRecyclingClass, 2250
AgsFxLowPassAudio, 2157
AgsFxLowPassAudioClass, 2157
AgsFxLowPassAudioProcessor, 2159
AgsFxLowPassAudioProcessorClass, 2159
AgsFxLowPassAudioSignal, 2161
AgsFxLowPassAudioSignalClass, 2162
AgsFxLowPassChannel, 2164
AgsFxLowPassChannel:enabled, 2164
AgsFxLowPassChannel:filter-gain, 2164
AgsFxLowPassChannel:q-lin, 2165
AgsFxLowPassChannelClass, 2164
AgsFxLowPassChannelProcessor, 2167
AgsFxLowPassChannelProcessorClass, 2167

AgsFxLowPassRecycling, 2169
AgsFxLowPassRecyclingClass, 2169
AgsFxLv2Audio, 2019
AgsFxLv2AudioChannelData, 2018
AgsFxLv2AudioClass, 2019
AgsFxLv2AudioFlags, 2018
AgsFxLv2AudioInputData, 2018
AgsFxLv2AudioProcessor, 2021
AgsFxLv2AudioProcessorClass, 2021
AgsFxLv2AudioScopeData, 2018
AgsFxLv2AudioSignal, 2023
AgsFxLv2AudioSignalClass, 2024
AgsFxLv2Channel, 2027
AgsFxLv2ChannelClass, 2027
AgsFxLv2ChannelInputData, 2027
AgsFxLv2ChannelProcessor, 2030
AgsFxLv2ChannelProcessorClass, 2030
AgsFxLv2Recycling, 2032
AgsFxLv2RecyclingClass, 2032
AgsFxNotationAudio, 1976
AgsFxNotationAudio:bpm, 1976
AgsFxNotationAudio:delay, 1976
AgsFxNotationAudio:duration, 1976
AgsFxNotationAudio:loop, 1976
AgsFxNotationAudio:loop-end, 1977
AgsFxNotationAudio:loop-start, 1977
AgsFxNotationAudio:tact, 1977
AgsFxNotationAudioClass, 1976
AgsFxNotationAudioFlags, 1975
AgsFxNotationAudioProcessor, 1981
AgsFxNotationAudioProcessorClass, 1981
AgsFxNotationAudioProcessorKeyMode, 1981
AgsFxNotationAudioSignal, 1984
AgsFxNotationAudioSignalClass, 1984
AgsFxNotationChannel, 1987
AgsFxNotationChannelClass, 1987
AgsFxNotationChannelProcessor, 1989
AgsFxNotationChannelProcessorClass, 1989
AgsFxNotationRecycling, 1992
AgsFxNotationRecyclingClass, 1992
AgsFxPatternAudio, 1954
AgsFxPatternAudio:bank-index-0, 1955
AgsFxPatternAudio:bank-index-1, 1955
AgsFxPatternAudio:bpm, 1955
AgsFxPatternAudio:delay, 1955
AgsFxPatternAudio:duration, 1956
AgsFxPatternAudio:loop, 1956
AgsFxPatternAudio:loop-end, 1956
AgsFxPatternAudio:loop-start, 1956
AgsFxPatternAudio:tact, 1956
AgsFxPatternAudioClass, 1955
AgsFxPatternAudioProcessor, 1959
AgsFxPatternAudioProcessorClass, 1959
AgsFxPatternAudioSignal, 1962
AgsFxPatternAudioSignalClass, 1962
AgsFxPatternChannel, 1965
AgsFxPatternChannelClass, 1965
AgsFxPatternChannelProcessor, 1967
AgsFxPatternChannelProcessorClass, 1967
AgsFxPatternRecycling, 1969
AgsFxPatternRecyclingClass, 1970
AgsFxPeakAudio, 2109
AgsFxPeakAudioClass, 2109
AgsFxPeakAudioProcessor, 2111
AgsFxPeakAudioProcessorClass, 2112
AgsFxPeakAudioSignal, 2114
AgsFxPeakAudioSignalClass, 2114
AgsFxPeakChannel, 2117
AgsFxPeakChannel:peak, 2117
AgsFxPeakChannelClass, 2117
AgsFxPeakChannelInputData, 2117
AgsFxPeakChannelProcessor, 2120
AgsFxPeakChannelProcessorClass, 2120
AgsFxPeakRecycling, 2122
AgsFxPeakRecyclingClass, 2122
AgsFxPlaybackAudio, 1918
AgsFxPlaybackAudio:bpm, 1918
AgsFxPlaybackAudio:capture-mode, 1919
AgsFxPlaybackAudio:delay, 1919
AgsFxPlaybackAudio:duration, 1919
AgsFxPlaybackAudio:export, 1919
AgsFxPlaybackAudio:file-audio-channels, 1919
AgsFxPlaybackAudio:file-buffer-size, 1920
AgsFxPlaybackAudio:file-format, 1920
AgsFxPlaybackAudio:file-samplerate, 1920
AgsFxPlaybackAudio:filename, 1920
AgsFxPlaybackAudio:loop, 1920
AgsFxPlaybackAudio:loop-end, 1921
AgsFxPlaybackAudio:loop-start, 1921
AgsFxPlaybackAudio:tact, 1921
AgsFxPlaybackAudioCaptureMode, 1917
AgsFxPlaybackAudioClass, 1918
AgsFxPlaybackAudioFlags, 1917
AgsFxPlaybackAudioProcessor, 1925
AgsFxPlaybackAudioProcessorClass, 1925
AgsFxPlaybackAudioProcessorDataMode, 1924
AgsFxPlaybackAudioSignal, 1927
AgsFxPlaybackAudioSignalClass, 1928
AgsFxPlaybackChannel, 1930
AgsFxPlaybackChannelClass, 1930
AgsFxPlaybackChannelProcessor, 1932
AgsFxPlaybackChannelProcessorClass, 1932
AgsFxPlaybackRecycling, 1934
AgsFxPlaybackRecyclingClass, 1935
AgsFxSF2SynthAudio, 2092
AgsFxSF2SynthAudio:chorus-delay, 2092
AgsFxSF2SynthAudio:chorus-depth, 2092
AgsFxSF2SynthAudio:chorus-enabled, 2093
AgsFxSF2SynthAudio:chorus-input-volume, 2093
AgsFxSF2SynthAudio:chorus-lfo-frequency, 2093
AgsFxSF2SynthAudio:chorus-lfo-oscillator, 2093
AgsFxSF2SynthAudio:chorus-mix, 2093
AgsFxSF2SynthAudio:chorus-output-volume, 2094
AgsFxSF2SynthAudio:chorus-pitch-type, 2094

AgsFxSF2SynthAudio:synth-key, 2094
AgsFxSF2SynthAudio:synth-octave, 2094
AgsFxSF2SynthAudio:synth-pitch-type, 2094
AgsFxSF2SynthAudio:synth-volume, 2095
AgsFxSF2SynthAudioChannelData, 2091
AgsFxSF2SynthAudioClass, 2092
AgsFxSF2SynthAudioInputData, 2092
AgsFxSF2SynthAudioProcessor, 2097
AgsFxSF2SynthAudioProcessorClass, 2097
AgsFxSF2SynthAudioScopeData, 2091
AgsFxSF2SynthAudioSignal, 2099
AgsFxSF2SynthAudioSignalClass, 2099
AgsFxSF2SynthChannel, 2101
AgsFxSF2SynthChannelClass, 2102
AgsFxSF2SynthChannelProcessor, 2104
AgsFxSF2SynthChannelProcessorClass, 2104
AgsFxSF2SynthRecycling, 2106
AgsFxSF2SynthRecyclingClass, 2106
AgsFxSynthAudio, 2038
AgsFxSynthAudio:chorus-delay, 2038
AgsFxSynthAudio:chorus-depth, 2038
AgsFxSynthAudio:chorus-enabled, 2039
AgsFxSynthAudio:chorus-input-volume, 2039
AgsFxSynthAudio:chorus-lfo-frequency, 2039
AgsFxSynthAudio:chorus-lfo-oscillator, 2039
AgsFxSynthAudio:chorus-mix, 2039
AgsFxSynthAudio:chorus-output-volume, 2040
AgsFxSynthAudio:chorus-pitch-type, 2040
AgsFxSynthAudio:high-pass-enabled, 2040
AgsFxSynthAudio:high-pass-filter-gain, 2040
AgsFxSynthAudio:high-pass-q-lin, 2040
AgsFxSynthAudio:low-pass-enabled, 2041
AgsFxSynthAudio:low-pass-filter-gain, 2041
AgsFxSynthAudio:low-pass-q-lin, 2041
AgsFxSynthAudio:noise-gain, 2041
AgsFxSynthAudio:pitch-tuning, 2041
AgsFxSynthAudio:pitch-type, 2042
AgsFxSynthAudio:sequencer-enabled, 2042
AgsFxSynthAudio:sequencer-sign, 2042
AgsFxSynthAudio:synth-0-key, 2042
AgsFxSynthAudio:synth-0-octave, 2042
AgsFxSynthAudio:synth-0-oscillator, 2043
AgsFxSynthAudio:synth-0-phase, 2043
AgsFxSynthAudio:synth-0-sync-attack-0, 2043
AgsFxSynthAudio:synth-0-sync-attack-1, 2043
AgsFxSynthAudio:synth-0-sync-attack-2, 2043
AgsFxSynthAudio:synth-0-sync-enabled, 2044
AgsFxSynthAudio:synth-0-sync-lfo-frequency, 2044
AgsFxSynthAudio:synth-0-sync-lfo-oscillator, 2044
AgsFxSynthAudio:synth-0-sync-phase-0, 2044
AgsFxSynthAudio:synth-0-sync-phase-1, 2044
AgsFxSynthAudio:synth-0-sync-phase-2, 2045
AgsFxSynthAudio:synth-0-sync-relative-attack-factor, 2045
AgsFxSynthAudio:synth-0-volume, 2045
AgsFxSynthAudio:synth-1-key, 2045
AgsFxSynthAudio:synth-1-octave, 2045
AgsFxSynthAudio:synth-1-oscillator, 2046
AgsFxSynthAudio:synth-1-phase, 2046
AgsFxSynthAudio:synth-1-sync-attack-0, 2046
AgsFxSynthAudio:synth-1-sync-attack-1, 2046
AgsFxSynthAudio:synth-1-sync-attack-2, 2046
AgsFxSynthAudio:synth-1-sync-enabled, 2047
AgsFxSynthAudio:synth-1-sync-lfo-frequency, 2047
AgsFxSynthAudio:synth-1-sync-lfo-oscillator, 2047
AgsFxSynthAudio:synth-1-sync-phase-0, 2047
AgsFxSynthAudio:synth-1-sync-phase-1, 2047
AgsFxSynthAudio:synth-1-sync-phase-2, 2048
AgsFxSynthAudio:synth-1-sync-relative-attack-factor, 2048
AgsFxSynthAudio:synth-1-volume, 2048
AgsFxSynthAudioChannelData, 2037
AgsFxSynthAudioClass, 2038
AgsFxSynthAudioInputData, 2038
AgsFxSynthAudioProcessor, 2050
AgsFxSynthAudioProcessorClass, 2050
AgsFxSynthAudioScopeData, 2037
AgsFxSynthAudioSignal, 2053
AgsFxSynthAudioSignalClass, 2053
AgsFxSynthChannel, 2055
AgsFxSynthChannelClass, 2055
AgsFxSynthChannelProcessor, 2057
AgsFxSynthChannelProcessorClass, 2057
AgsFxSynthRecycling, 2060
AgsFxSynthRecyclingClass, 2060
AgsFxTwoPassAliaseAudio, 2252
AgsFxTwoPassAliaseAudioClass, 2252
AgsFxTwoPassAliaseAudioProcessor, 2254
AgsFxTwoPassAliaseAudioProcessorClass, 2254
AgsFxTwoPassAliaseAudioSignal, 2256
AgsFxTwoPassAliaseAudioSignalClass, 2257
AgsFxTwoPassAliaseChannel, 2263
AgsFxTwoPassAliaseChannel:a-amount, 2264
AgsFxTwoPassAliaseChannel:a-phase, 2264
AgsFxTwoPassAliaseChannel:b-amount, 2264
AgsFxTwoPassAliaseChannel:b-phase, 2264
AgsFxTwoPassAliaseChannel:enabled, 2264
AgsFxTwoPassAliaseChannelClass, 2263
AgsFxTwoPassAliaseChannelInputData, 2263
AgsFxTwoPassAliaseChannelProcessor, 2267
AgsFxTwoPassAliaseChannelProcessorClass, 2267
AgsFxTwoPassAliaseRecycling, 2269
AgsFxTwoPassAliaseRecyclingClass, 2269
AgsFxVolumeAudio, 2142
AgsFxVolumeAudio:muted, 2142
AgsFxVolumeAudioClass, 2142
AgsFxVolumeAudioProcessor, 2145
AgsFxVolumeAudioProcessorClass, 2145
AgsFxVolumeAudioSignal, 2147
AgsFxVolumeAudioSignalClass, 2147
AgsFxVolumeChannel, 2149
AgsFxVolumeChannel:muted, 2150
AgsFxVolumeChannel:volume, 2150
AgsFxVolumeChannelClass, 2149
AgsFxVolumeChannelProcessor, 2152
AgsFxVolumeChannelProcessorClass, 2152

AgsFxVolumeRecycling, 2154
AgsFxVolumeRecyclingClass, 2155

G

AGS_GENERIC_RECALL_CHANNEL_RUN, 1445
AGS_GENERIC_RECALL_CHANNEL_RUN_CLASS, 1445
AGS_GENERIC_RECALL_CHANNEL_RUN_GET_CLASS, 1445
ags_generic_recall_channel_run_get_type, 1445
ags_generic_recall_channel_run_new, 1444
AGS_GENERIC_RECALL_RECYCLING, 1447
AGS_GENERIC_RECALL_RECYCLING_CLASS, 1447
AGS_GENERIC_RECALL_RECYCLING_GET_CLASS, 1448
ags_generic_recall_recycling_get_type, 1448
ags_generic_recall_recycling_new, 1447
AgsGenericRecallChannelRun, 1446
AgsGenericRecallChannelRun:generic-recall-recycling-child-type, 1446
AgsGenericRecallChannelRunClass, 1446
AgsGenericRecallRecycling, 1448
AgsGenericRecallRecyclingClass, 1448
AgsGigLevel, 1837
AGS_GSTREAMER_CLIENT, 826
ags_gstreamer_client_activate, 825
ags_gstreamer_client_add_device, 824
ags_gstreamer_client_add_port, 824
AGS_GSTREAMER_CLIENT_CLASS, 826
ags_gstreamer_client_close, 823
ags_gstreamer_client_deactivate, 825
ags_gstreamer_client_find, 823
ags_gstreamer_client_find_uuid, 822
ags_gstreamer_client_flags_get_type, 826
AGS_GSTREAMER_CLIENT_GET_CLASS, 826
AGS_GSTREAMER_CLIENT_GET_OBJ_MUTEX, 821
ags_gstreamer_client_get_type, 826
ags_gstreamer_client_new, 825
ags_gstreamer_client_open, 823
ags_gstreamer_client_remove_device, 824
ags_gstreamer_client_remove_port, 824
ags_gstreamer_client_set_flags, 822
ags_gstreamer_client_test_flags, 821
ags_gstreamer_client_unset_flags, 822
AGS_GSTREAMER_DEVIN, 839
ags_gstreamer_devin_adjust_delay_and_attack, 839
AGS_GSTREAMER_DEVIN_CLASS, 839
AGS_GSTREAMER_DEVIN_DEFAULT_APP_BUFFER_SIZE, 840
AGS_GSTREAMER_DEVIN_ERROR, 842
ags_gstreamer_devin_error_quark, 837
ags_gstreamer_devin_flags_get_type, 840
AGS_GSTREAMER_DEVIN_GET_CLASS, 840
AGS_GSTREAMER_DEVIN_GET_OBJ_MUTEX, 837
ags_gstreamer_devin_get_type, 840
ags_gstreamer_devin_new, 839
ags_gstreamer_devin_realloc_buffer, 839
ags_gstreamer_devin_set_flags, 838
ags_gstreamer_devin_switch_buffer_flag, 838
ags_gstreamer_devin_test_flags, 837
ags_gstreamer_devin_unset_flags, 838
AGS_GSTREAMER_DEVOUT, 850
ags_gstreamer_devout_adjust_delay_and_attack, 849
AGS_GSTREAMER_DEVOUT_CLASS, 850
AGS_GSTREAMER_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 850
AGS_GSTREAMER_DEVOUT_ERROR, 853
ags_gstreamer_devout_error_quark, 847
ags_gstreamer_devout_flags_get_type, 850
AGS_GSTREAMER_DEVOUT_GET_CLASS, 850
AGS_GSTREAMER_DEVOUT_GET_OBJ_MUTEX, 847
ags_gstreamer_devout_get_type, 850
ags_gstreamer_devout_new, 849
ags_gstreamer_devout_realloc_buffer, 849
ags_gstreamer_devout_set_flags, 848
ags_gstreamer_devout_switch_buffer_flag, 849
ags_gstreamer_devout_test_flags, 848
ags_gstreamer_devout_unset_flags, 848
AGS_GSTREAMER_FILE, 1824
ags_gstreamer_file_check_suffix, 1821
AGS_GSTREAMER_FILE_CHUNK_FRAME_COUNT, 1820
AGS_GSTREAMER_FILE_CLASS, 1824
ags_gstreamer_file_create_aac_encoding_profile, 1822
ags_gstreamer_file_create_mkv_encoding_profile, 1823
ags_gstreamer_file_create_mp3_encoding_profile, 1822
ags_gstreamer_file_create_mp4_encoding_profile, 1822
ags_gstreamer_file_create_mpeg_encoding_profile, 1823
ags_gstreamer_file_create_wav_encoding_profile, 1821
ags_gstreamer_file_create_webm_encoding_profile, 1823
AGS_GSTREAMER_FILE_DEFAULT_AUDIO_CHANNELS, 1825
AGS_GSTREAMER_FILE_DEFAULT_DISCOVERER_TIMEOU 1825
AGS_GSTREAMER_FILE_DEFAULT_FLUSH_IDLE, 1825
AGS_GSTREAMER_FILE_DEFAULT_FORMAT, 1825
ags_gstreamer_file_detect_encoding_profile, 1824
AGS_GSTREAMER_FILE_GET_CLASS, 1824
AGS_GSTREAMER_FILE_GET_OBJ_MUTEX, 1820
ags_gstreamer_file_get_type, 1825
ags_gstreamer_file_new, 1824
ags_gstreamer_file_set_flags, 1820
ags_gstreamer_file_test_flags, 1820
ags_gstreamer_file_unset_flags, 1821
AGS_GSTREAMER_PORT, 833
AGS_GSTREAMER_PORT_CLASS, 833
AGS_GSTREAMER_PORT_DEFAULT_CACHE_BUFFER_SIZE 834
ags_gstreamer_port_find, 830
ags_gstreamer_port_flags_get_type, 833
AGS_GSTREAMER_PORT_GET_CLASS, 833
ags_gstreamer_port_get_fixed_size, 831
ags_gstreamer_port_get_latency, 832
AGS_GSTREAMER_PORT_GET_OBJ_MUTEX, 829
ags_gstreamer_port_get_type, 833
ags_gstreamer_port_new, 832
ags_gstreamer_port_register, 831

ags_gstreamer_port_set_buffer_size, 832
ags_gstreamer_port_set_cache_buffer_size, 832
ags_gstreamer_port_set_flags, 830
ags_gstreamer_port_set_format, 832
ags_gstreamer_port_set_pcm_channels, 832
ags_gstreamer_port_set_samplerate, 831
ags_gstreamer_port_test_flags, 829
ags_gstreamer_port_unregister, 831
ags_gstreamer_port_unset_flags, 830
AGS_GSTREAMER_SERVER, 818
ags_gstreamer_server_add_client, 816
AGS_GSTREAMER_SERVER_CLASS, 818
ags_gstreamer_server_connect_client, 817
ags_gstreamer_server_disconnect_client, 817
ags_gstreamer_server_find_client, 815
ags_gstreamer_server_find_port, 816
ags_gstreamer_server_find_url, 815
AGS_GSTREAMER_SERVER_GET_CLASS, 818
AGS_GSTREAMER_SERVER_GET_OBJ_MUTEX, 814
ags_gstreamer_server_get_type, 818
ags_gstreamer_server_new, 818
ags_gstreamer_server_remove_client, 817
ags_gstreamer_server_set_flags, 814
ags_gstreamer_server_start_poll, 817
ags_gstreamer_server_test_flags, 814
ags_gstreamer_server_unset_flags, 815
AgsGstreamerClient, 827
AgsGstreamerClient:client-name, 827
AgsGstreamerClient:device, 827
AgsGstreamerClient:gstreamer-server, 828
AgsGstreamerClient:port, 828
AgsGstreamerClientClass, 827
AgsGstreamerClientFlags, 826
AgsGstreamerDevin, 843
AgsGstreamerDevin:attack, 843
AgsGstreamerDevin:bpm, 843
AgsGstreamerDevin:buffer, 843
AgsGstreamerDevin:buffer-size, 844
AgsGstreamerDevin:delay-factor, 844
AgsGstreamerDevin:device, 844
AgsGstreamerDevin:dsp-channels, 844
AgsGstreamerDevin:format, 845
AgsGstreamerDevin:gstreamer-client, 845
AgsGstreamerDevin:gstreamer-port, 845
AgsGstreamerDevin:pcm-channels, 845
AgsGstreamerDevin:samplerate, 846
AgsGstreamerDevinAppBufferMode, 841
AgsGstreamerDevinClass, 843
AgsGstreamerDevinError, 842
AgsGstreamerDevinFlags, 840
AgsGstreamerDevinSyncFlags, 841
AgsGstreamerDevout, 853
AgsGstreamerDevout:attack, 853
AgsGstreamerDevout:bpm, 854
AgsGstreamerDevout:buffer, 854
AgsGstreamerDevout:buffer-size, 854
AgsGstreamerDevout:delay-factor, 854
AgsGstreamerDevout:device, 855
AgsGstreamerDevout:dsp-channels, 855
AgsGstreamerDevout:format, 855
AgsGstreamerDevout:gstreamer-client, 855
AgsGstreamerDevout:gstreamer-port, 855
AgsGstreamerDevout:pcm-channels, 856
AgsGstreamerDevout:samplerate, 856
AgsGstreamerDevoutAppBufferMode, 851
AgsGstreamerDevoutClass, 853
AgsGstreamerDevoutError, 853
AgsGstreamerDevoutFlags, 851
AgsGstreamerDevoutSyncFlags, 852
AgsGstreamerFile, 1825
AgsGstreamerFile:audio-channels, 1826
AgsGstreamerFile:buffer-size, 1826
AgsGstreamerFile:filename, 1826
AgsGstreamerFile:format, 1826
AgsGstreamerFile:samplerate, 1827
AgsGstreamerFileClass, 1825
AgsGstreamerPort, 834
AgsGstreamerPort:gstreamer-client, 835
AgsGstreamerPort:gstreamer-devin, 835
AgsGstreamerPort:gstreamer-devout, 835
AgsGstreamerPort:port-name, 835
AgsGstreamerPortClass, 835
AgsGstreamerPortFlags, 834
AgsGstreamerServer, 819
AgsGstreamerServer:default-gstreamer-client, 819
AgsGstreamerServer:default-soundcard, 819
AgsGstreamerServer:gstreamer-client, 819
AgsGstreamerServer:url, 820
AgsGstreamerServerClass, 819

H

AGS_HQ_PITCH_UTIL, 231
ags_hq_pitch_util_alloc, 223
ags_hq_pitch_util_copy, 223
AGS_HQ_PITCH_UTIL_DEFAULT_MAX_BUFFER_SIZE, 232
ags_hq_pitch_util_free, 223
ags_hq_pitch_util_get_base_key, 228
ags_hq_pitch_util_get_buffer_length, 226
ags_hq_pitch_util_get_destination, 225
ags_hq_pitch_util_get_destination_stride, 225
ags_hq_pitch_util_get_format, 227
ags_hq_pitch_util_get_samplerate, 227
ags_hq_pitch_util_get_source, 223
ags_hq_pitch_util_get_source_stride, 224
ags_hq_pitch_util_get_tuning, 229
ags_hq_pitch_util_get_type, 232
ags_hq_pitch_util_pitch, 231
ags_hq_pitch_util_pitch_complex, 231
ags_hq_pitch_util_pitch_double, 231
ags_hq_pitch_util_pitch_float, 230
ags_hq_pitch_util_pitch_s16, 229
ags_hq_pitch_util_pitch_s24, 230
ags_hq_pitch_util_pitch_s32, 230

ags_hq_pitch_util_pitch_s64, 230
ags_hq_pitch_util_pitch_s8, 229
ags_hq_pitch_util_set_base_key, 228
ags_hq_pitch_util_set_buffer_length, 226
ags_hq_pitch_util_set_destination, 225
ags_hq_pitch_util_set_destination_stride, 226
ags_hq_pitch_util_set_format, 227
ags_hq_pitch_util_set_samplerate, 228
ags_hq_pitch_util_set_source, 224
ags_hq_pitch_util_set_source_stride, 224
ags_hq_pitch_util_set_tuning, 229
AgsHQPitchUtil, 232

I

AGS_INPUT, 1103
AgsInput, 1104
AgsInput:file-link, 1104
AgsInput:synth-generator, 1104
ags_input_add_synth_generator, 1101
AGS_INPUT_CLASS, 1103
AGS_INPUT_GET_CLASS, 1103
ags_input_get_type, 1103
ags_input_is_active, 1101
ags_input_new, 1103
ags_input_next_active, 1101
ags_input_open_file, 1102
ags_input_remove_synth_generator, 1102
AGS_INPUT_SYNTH_BASE_NOTE, 1104
AgsInputClass, 1104
AGS_IPATCH, 1830
AgsIpatch, 1831
AgsIpatch:filename, 1831
AgsIpatch:mode, 1832
AgsIpatch:soundcard, 1832
ags_ipatch_check_suffix, 1829
AGS_IPATCH_CLASS, 1830
AGS_IPATCH_DEFAULT_CHANNELS, 1830
AGS_IPATCH_DLS2_READER, 1851
AGS_IPATCH_DLS2_READER_CLASS, 1851
AGS_IPATCH_DLS2_READER_GET_CLASS, 1852
ags_ipatch_dls2_reader_get_instrument_all, 1850
AGS_IPATCH_DLS2_READER_GET_OBJ_MUTEX, 1848
ags_ipatch_dls2_reader_get_sample_all, 1850
ags_ipatch_dls2_reader_get_sample_by_instrument_index, 1851
ags_ipatch_dls2_reader_get_type, 1852
ags_ipatch_dls2_reader_load, 1849
ags_ipatch_dls2_reader_new, 1851
ags_ipatch_dls2_reader_select_instrument, 1849
ags_ipatch_dls2_reader_select_sample, 1849
ags_ipatch_dls2_reader_set_flags, 1848
ags_ipatch_dls2_reader_test_flags, 1848
ags_ipatch_dls2_reader_unset_flags, 1848
ags_ipatch_flags_get_type, 1830
AGS_IPATCH_GET_CLASS, 1830
AGS_IPATCH_GET_OBJ_MUTEX, 1828
ags_ipatch_get_type, 1830
AGS_IPATCH_GIG_READER, 1837
AGS_IPATCH_GIG_READER_CLASS, 1837
AGS_IPATCH_GIG_READER_GET_CLASS, 1837
ags_ipatch_gig_reader_get_instrument_all, 1835
AGS_IPATCH_GIG_READER_GET_OBJ_MUTEX, 1833
ags_ipatch_gig_reader_get_sample_all, 1836
ags_ipatch_gig_reader_get_sample_by_instrument_index, 1836
ags_ipatch_gig_reader_get_type, 1837
ags_ipatch_gig_reader_load, 1834
ags_ipatch_gig_reader_new, 1836
ags_ipatch_gig_reader_select_instrument, 1835
ags_ipatch_gig_reader_select_sample, 1835
ags_ipatch_gig_reader_set_flags, 1834
ags_ipatch_gig_reader_test_flags, 1833
ags_ipatch_gig_reader_unset_flags, 1834
ags_ipatch_new, 1829
AGS_IPATCH_READ, 1831
AGS_IPATCH_SAMPLE, 1855
AGS_IPATCH_SAMPLE_CLASS, 1855
AGS_IPATCH_SAMPLE_GET_CLASS, 1855
AGS_IPATCH_SAMPLE_GET_OBJ_MUTEX, 1854
ags_ipatch_sample_get_type, 1856
ags_ipatch_sample_new, 1855
ags_ipatch_sample_set_flags, 1854
ags_ipatch_sample_test_flags, 1854
ags_ipatch_sample_unset_flags, 1855
ags_ipatch_set_flags, 1829
AGS_IPATCH_SF2_READER, 1845
AGS_IPATCH_SF2_READER_CLASS, 1845
AGS_IPATCH_SF2_READER_GET_CLASS, 1845
ags_ipatch_sf2_reader_get_instrument_all, 1842
ags_ipatch_sf2_reader_get_instrument_by_preset_index, 1843
AGS_IPATCH_SF2_READER_GET_OBJ_MUTEX, 1840
ags_ipatch_sf2_reader_get_preset_all, 1842
ags_ipatch_sf2_reader_get_sample_all, 1843
ags_ipatch_sf2_reader_get_sample_by_preset_and_instrument_inc
1844
ags_ipatch_sf2_reader_get_sample_by_preset_index, 1844
ags_ipatch_sf2_reader_get_type, 1845
ags_ipatch_sf2_reader_load, 1841
ags_ipatch_sf2_reader_new, 1844
ags_ipatch_sf2_reader_select_instrument, 1841
ags_ipatch_sf2_reader_select_preset, 1841
ags_ipatch_sf2_reader_select_sample, 1842
ags_ipatch_sf2_reader_set_flags, 1840
ags_ipatch_sf2_reader_test_flags, 1840
ags_ipatch_sf2_reader_unset_flags, 1840
ags_ipatch_test_flags, 1828
ags_ipatch_unset_flags, 1829
AGS_IPATCH_WRITE, 1831
AgsIpatchClass, 1831
AgsIpatchDLS2Reader, 1852
AgsIpatchDLS2Reader:ipatch, 1853
AgsIpatchDLS2ReaderClass, 1852
AgsIpatchFlags, 1831
AgsIpatchGigReader, 1838
AgsIpatchGigReader:ipatch, 1838
AgsIpatchGigReaderClass, 1838

AgsIpatchSample, 1856
AgsIpatchSample:buffer-size, 1856
AgsIpatchSample:format, 1856
AgsIpatchSample:sample, 1857
AgsIpatchSampleClass, 1856
AgsIpatchSF2Reader, 1846
AgsIpatchSF2Reader:ipatch, 1846
AgsIpatchSF2ReaderClass, 1846
AGS_IS_ACCELERATION, 1242
AGS_IS_ACCELERATION_CLASS, 1242
AGS_IS_ADD_AUDIO, 2272
AGS_IS_ADD_AUDIO_CLASS, 2272
AGS_IS_ADD_AUDIO_SIGNAL, 2274
AGS_IS_ADD_AUDIO_SIGNAL_CLASS, 2275
AGS_IS_ADD_NOTE, 2278
AGS_IS_ADD_NOTE_CLASS, 2278
AGS_IS_ADD_SOUNDCARD, 2281
AGS_IS_ADD_SOUNDCARD_CLASS, 2281
AGS_IS_ALSA_DEVIN, 649
AGS_IS_ALSA_DEVIN_CLASS, 649
AGS_IS_ALSA_DEVOUT, 659
AGS_IS_ALSA_DEVOUT_CLASS, 659
AGS_IS_ALSA_MIDIIN, 668
AGS_IS_ALSA_MIDIIN_CLASS, 668
AGS_IS_APPLY_BPM, 2284
AGS_IS_APPLY_BPM_CLASS, 2284
AGS_IS_APPLY_PRESETS, 2287
AGS_IS_APPLY_PRESETS_CLASS, 2287
AGS_IS_APPLY_SEQUENCER_LENGTH, 2290
AGS_IS_APPLY_SEQUENCER_LENGTH_CLASS, 2290
AGS_IS_APPLY_SF2_MIDI_LOCALE, 2303
AGS_IS_APPLY_SF2_MIDI_LOCALE_CLASS, 2303
AGS_IS_APPLY_SF2_SYNTH, 2300
AGS_IS_APPLY_SF2_SYNTH_CLASS, 2300
AGS_IS_APPLY_SFZ_INSTRUMENT, 2309
AGS_IS_APPLY_SFZ_INSTRUMENT_CLASS, 2309
AGS_IS_APPLY_SFZ_SYNTH, 2306
AGS_IS_APPLY_SFZ_SYNTH_CLASS, 2306
AGS_IS_APPLY_SOUND_CONFIG, 2293
AGS_IS_APPLY_SOUND_CONFIG_CLASS, 2293
AGS_IS_APPLY_SYNTH, 2296
AGS_IS_APPLY_SYNTH_CLASS, 2296
AGS_IS_APPLY_TACT, 2312
AGS_IS_APPLY_TACT_CLASS, 2312
AGS_IS_AUDIO, 1020
AGS_IS_AUDIO_APPLICATION_CONTEXT, 957
AGS_IS_AUDIO_APPLICATION_CONTEXT_CLASS, 957
AGS_IS_AUDIO_CLASS, 1020
AGS_IS_AUDIO_CONTAINER, 1789
AGS_IS_AUDIO_CONTAINER_CLASS, 1789
AGS_IS_AUDIO_CONTAINER_MANAGER, 1780
AGS_IS_AUDIO_CONTAINER_MANAGER_CLASS, 1780
AGS_IS_AUDIO_FILE, 1812
AGS_IS_AUDIO_FILE_CLASS, 1812
AGS_IS_AUDIO_FILE_LINK, 1817
AGS_IS_AUDIO_FILE_LINK_CLASS, 1817
AGS_IS_AUDIO_FILE_MANAGER, 1796
AGS_IS_AUDIO_FILE_MANAGER_CLASS, 1796
AGS_IS_AUDIO_LOOP, 1454
AGS_IS_AUDIO_LOOP_CLASS, 1454
AGS_IS_AUDIO_SIGNAL, 1141
AGS_IS_AUDIO_SIGNAL_CLASS, 1141
AGS_IS_AUDIO_THREAD, 1462
AGS_IS_AUDIO_THREAD_CLASS, 1463
AGS_IS_AUDIO_UNIT_CLIENT, 918
AGS_IS_AUDIO_UNIT_CLIENT_CLASS, 918
AGS_IS_AUDIO_UNIT_DEVIN, 931
AGS_IS_AUDIO_UNIT_DEVIN_CLASS, 931
AGS_IS_AUDIO_UNIT_DEVOUT, 941
AGS_IS_AUDIO_UNIT_DEVOUT_CLASS, 941
AGS_IS_AUDIO_UNIT_PORT, 925
AGS_IS_AUDIO_UNIT_PORT_CLASS, 925
AGS_IS_AUDIO_UNIT_SERVER, 911
AGS_IS_AUDIO_UNIT_SERVER_CLASS, 911
AGS_IS_AUTOMATION, 1231
AGS_IS_AUTOMATION_CLASS, 1231
AGS_IS_BASE_PLUGIN, 18
AGS_IS_BASE_PLUGIN_CLASS, 18
AGS_IS_BUFFER, 1271
AGS_IS_BUFFER_CLASS, 1271
AGS_IS_CANCEL_AUDIO, 2315
AGS_IS_CANCEL_AUDIO_CLASS, 2315
AGS_IS_CANCEL_CHANNEL, 2318
AGS_IS_CANCEL_CHANNEL_CLASS, 2318
AGS_IS_CHANNEL, 1081
AGS_IS_CHANNEL_CLASS, 1081
AGS_IS_CHANNEL_THREAD, 1471
AGS_IS_CHANNEL_THREAD_CLASS, 1471
AGS_IS_CLEAR_AUDIO_SIGNAL, 2321
AGS_IS_CLEAR_AUDIO_SIGNAL_CLASS, 2321
AGS_IS_CLEAR_BUFFER, 2324
AGS_IS_CLEAR_BUFFER_CLASS, 2324
AGS_IS_CORE_AUDIO_CLIENT, 869
AGS_IS_CORE_AUDIO_CLIENT_CLASS, 869
AGS_IS_CORE_AUDIO_DEVIN, 882
AGS_IS_CORE_AUDIO_DEVIN_CLASS, 882
AGS_IS_CORE_AUDIO_DEVOUT, 892
AGS_IS_CORE_AUDIO_DEVOUT_CLASS, 892
AGS_IS_CORE_AUDIO_MIDIIN, 901
AGS_IS_CORE_AUDIO_MIDIIN_CLASS, 901
AGS_IS_CORE_AUDIO_PORT, 876
AGS_IS_CORE_AUDIO_PORT_CLASS, 876
AGS_IS_CORE_AUDIO_SERVER, 861
AGS_IS_CORE_AUDIO_SERVER_CLASS, 861
AGS_IS_CROP_NOTE, 2327
AGS_IS_CROP_NOTE_CLASS, 2327
AGS_IS_DSSI_MANAGER, 62
AGS_IS_DSSI_MANAGER_CLASS, 62
AGS_IS_DSSI_PLUGIN, 55
AGS_IS_DSSI_PLUGIN_CLASS, 55
AGS_IS_EXPORT_OUTPUT, 2331
AGS_IS_EXPORT_OUTPUT_CLASS, 2332
AGS_IS_EXPORT_THREAD, 1476
AGS_IS_EXPORT_THREAD_CLASS, 1476

AGS_IS_FIFOOUT, 640
AGS_IS_FIFOOUT_CLASS, 640
AGS_IS_FREE_SELECTION, 2335
AGS_IS_FREE_SELECTION_CLASS, 2335
AGS_IS_FREQUENCY_MAP, 631
AGS_IS_FREQUENCY_MAP_CLASS, 631
AGS_IS_FREQUENCY_MAP_MANAGER, 626
AGS_IS_FREQUENCY_MAP_MANAGER_CLASS, 626
AGS_IS_FX_ANALYSE_AUDIO, 2124
AGS_IS_FX_ANALYSE_AUDIO_CLASS, 2124
AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR, 2126
AGS_IS_FX_ANALYSE_AUDIO_PROCESSOR_CLASS, 2126
AGS_IS_FX_ANALYSE_AUDIO_SIGNAL, 2128
AGS_IS_FX_ANALYSE_AUDIO_SIGNAL_CLASS, 2128
AGS_IS_FX_ANALYSE_CHANNEL, 2133
AGS_IS_FX_ANALYSE_CHANNEL_CLASS, 2133
AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR, 2136
AGS_IS_FX_ANALYSE_CHANNEL_PROCESSOR_CLASS, 2136
AGS_IS_FX_ANALYSE_RECYCLING, 2139
AGS_IS_FX_ANALYSE_RECYCLING_CLASS, 2139
AGS_IS_FX_BUFFER_AUDIO, 1936
AGS_IS_FX_BUFFER_AUDIO_CLASS, 1936
AGS_IS_FX_BUFFER_AUDIO_PROCESSOR, 1939
AGS_IS_FX_BUFFER_AUDIO_PROCESSOR_CLASS, 1939
AGS_IS_FX_BUFFER_AUDIO_SIGNAL, 1941
AGS_IS_FX_BUFFER_AUDIO_SIGNAL_CLASS, 1941
AGS_IS_FX_BUFFER_CHANNEL, 1946
AGS_IS_FX_BUFFER_CHANNEL_CLASS, 1946
AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR, 1948
AGS_IS_FX_BUFFER_CHANNEL_PROCESSOR_CLASS, 1948
AGS_IS_FX_BUFFER_RECYCLING, 1951
AGS_IS_FX_BUFFER_RECYCLING_CLASS, 1951
AGS_IS_FX_DSSI_AUDIO, 1998
AGS_IS_FX_DSSI_AUDIO_CLASS, 1998
AGS_IS_FX_DSSI_AUDIO_PROCESSOR, 2001
AGS_IS_FX_DSSI_AUDIO_PROCESSOR_CLASS, 2001
AGS_IS_FX_DSSI_AUDIO_SIGNAL, 2003
AGS_IS_FX_DSSI_AUDIO_SIGNAL_CLASS, 2004
AGS_IS_FX_DSSI_CHANNEL, 2006
AGS_IS_FX_DSSI_CHANNEL_CLASS, 2006
AGS_IS_FX_DSSI_CHANNEL_PROCESSOR, 2008
AGS_IS_FX_DSSI_CHANNEL_PROCESSOR_CLASS, 2009
AGS_IS_FX_DSSI_RECYCLING, 2011
AGS_IS_FX_DSSI_RECYCLING_CLASS, 2011
AGS_IS_FX_ENVELOPE_AUDIO, 2203
AGS_IS_FX_ENVELOPE_AUDIO_CLASS, 2203
AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR, 2206
AGS_IS_FX_ENVELOPE_AUDIO_PROCESSOR_CLASS, 2206
AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL, 2208
AGS_IS_FX_ENVELOPE_AUDIO_SIGNAL_CLASS, 2208
AGS_IS_FX_ENVELOPE_CHANNEL, 2211
AGS_IS_FX_ENVELOPE_CHANNEL_CLASS, 2211
AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR, 2214
AGS_IS_FX_ENVELOPE_CHANNEL_PROCESSOR_CLASS, 2214
AGS_IS_FX_ENVELOPE_RECYCLING, 2216
AGS_IS_FX_ENVELOPE_RECYCLING_CLASS, 2216
AGS_IS_FX_EQ10_AUDIO, 2186
AGS_IS_FX_EQ10_AUDIO_CLASS, 2186
AGS_IS_FX_EQ10_AUDIO_PROCESSOR, 2188
AGS_IS_FX_EQ10_AUDIO_PROCESSOR_CLASS, 2188
AGS_IS_FX_EQ10_AUDIO_SIGNAL, 2190
AGS_IS_FX_EQ10_AUDIO_SIGNAL_CLASS, 2190
AGS_IS_FX_EQ10_CHANNEL, 2194
AGS_IS_FX_EQ10_CHANNEL_CLASS, 2194
AGS_IS_FX_EQ10_CHANNEL_PROCESSOR, 2199
AGS_IS_FX_EQ10_CHANNEL_PROCESSOR_CLASS, 2199
AGS_IS_FX_EQ10_RECYCLING, 2201
AGS_IS_FX_EQ10_RECYCLING_CLASS, 2201
AGS_IS_FX_FACTORY, 1320
AGS_IS_FX_FACTORY_CLASS, 1320
AGS_IS_FX_FM_SYNTH_AUDIO, 2064
AGS_IS_FX_FM_SYNTH_AUDIO_CLASS, 2064
AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR, 2077
AGS_IS_FX_FM_SYNTH_AUDIO_PROCESSOR_CLASS, 2077
AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL, 2079
AGS_IS_FX_FM_SYNTH_AUDIO_SIGNAL_CLASS, 2079
AGS_IS_FX_FM_SYNTH_CHANNEL, 2081
AGS_IS_FX_FM_SYNTH_CHANNEL_CLASS, 2082
AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR, 2084
AGS_IS_FX_FM_SYNTH_CHANNEL_PROCESSOR_CLASS, 2084
AGS_IS_FX_FM_SYNTH_RECYCLING, 2086
AGS_IS_FX_FM_SYNTH_RECYCLING_CLASS, 2086
AGS_IS_FX_HIGH_PASS_AUDIO, 2171
AGS_IS_FX_HIGH_PASS_AUDIO_CLASS, 2171
AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR, 2173
AGS_IS_FX_HIGH_PASS_AUDIO_PROCESSOR_CLASS, 2173
AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL, 2176
AGS_IS_FX_HIGH_PASS_AUDIO_SIGNAL_CLASS, 2176
AGS_IS_FX_HIGH_PASS_CHANNEL, 2178
AGS_IS_FX_HIGH_PASS_CHANNEL_CLASS, 2178
AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR, 2181
AGS_IS_FX_HIGH_PASS_CHANNEL_PROCESSOR_CLASS, 2181
AGS_IS_FX_HIGH_PASS_RECYCLING, 2183
AGS_IS_FX_HIGH_PASS_RECYCLING_CLASS, 2183
AGS_IS_FX_LADSPA_AUDIO, 2220
AGS_IS_FX_LADSPA_AUDIO_CLASS, 2220
AGS_IS_FX_LADSPA_AUDIO_PROCESSOR, 2222
AGS_IS_FX_LADSPA_AUDIO_PROCESSOR_CLASS, 2222
AGS_IS_FX_LADSPA_AUDIO_SIGNAL, 2224
AGS_IS_FX_LADSPA_AUDIO_SIGNAL_CLASS, 2224
AGS_IS_FX_LADSPA_CHANNEL, 2228
AGS_IS_FX_LADSPA_CHANNEL_CLASS, 2228
AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR, 2230
AGS_IS_FX_LADSPA_CHANNEL_PROCESSOR_CLASS, 2230

AGS_IS_FX_LADSPA_RECYCLING, 2233
AGS_IS_FX_LADSPA_RECYCLING_CLASS, 2233
AGS_IS_FX_LFO_AUDIO, 2235
AGS_IS_FX_LFO_AUDIO_CLASS, 2235
AGS_IS_FX_LFO_AUDIO_PROCESSOR, 2237
AGS_IS_FX_LFO_AUDIO_PROCESSOR_CLASS, 2237
AGS_IS_FX_LFO_AUDIO_SIGNAL, 2240
AGS_IS_FX_LFO_AUDIO_SIGNAL_CLASS, 2240
AGS_IS_FX_LFO_CHANNEL, 2243
AGS_IS_FX_LFO_CHANNEL_CLASS, 2243
AGS_IS_FX_LFO_CHANNEL_PROCESSOR, 2247
AGS_IS_FX_LFO_CHANNEL_PROCESSOR_CLASS, 2247
AGS_IS_FX_LFO_RECYCLING, 2249
AGS_IS_FX_LFO_RECYCLING_CLASS, 2249
AGS_IS_FX_LOW_PASS_AUDIO, 2156
AGS_IS_FX_LOW_PASS_AUDIO_CLASS, 2156
AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR, 2159
AGS_IS_FX_LOW_PASS_AUDIO_PROCESSOR_CLASS, 2159
AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL, 2161
AGS_IS_FX_LOW_PASS_AUDIO_SIGNAL_CLASS, 2161
AGS_IS_FX_LOW_PASS_CHANNEL, 2163
AGS_IS_FX_LOW_PASS_CHANNEL_CLASS, 2164
AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR, 2166
AGS_IS_FX_LOW_PASS_CHANNEL_PROCESSOR_CLASS, 2166
AGS_IS_FX_LOW_PASS_RECYCLING, 2169
AGS_IS_FX_LOW_PASS_RECYCLING_CLASS, 2169
AGS_IS_FX_LV2_AUDIO, 2017
AGS_IS_FX_LV2_AUDIO_CLASS, 2017
AGS_IS_FX_LV2_AUDIO_PROCESSOR, 2021
AGS_IS_FX_LV2_AUDIO_PROCESSOR_CLASS, 2021
AGS_IS_FX_LV2_AUDIO_SIGNAL, 2023
AGS_IS_FX_LV2_AUDIO_SIGNAL_CLASS, 2023
AGS_IS_FX_LV2_CHANNEL, 2027
AGS_IS_FX_LV2_CHANNEL_CLASS, 2027
AGS_IS_FX_LV2_CHANNEL_PROCESSOR, 2029
AGS_IS_FX_LV2_CHANNEL_PROCESSOR_CLASS, 2029
AGS_IS_FX_LV2_RECYCLING, 2031
AGS_IS_FX_LV2_RECYCLING_CLASS, 2032
AGS_IS_FX_NOTATION_AUDIO, 1975
AGS_IS_FX_NOTATION_AUDIO_CLASS, 1975
AGS_IS_FX_NOTATION_AUDIO_PROCESSOR, 1980
AGS_IS_FX_NOTATION_AUDIO_PROCESSOR_CLASS, 1980
AGS_IS_FX_NOTATION_AUDIO_SIGNAL, 1984
AGS_IS_FX_NOTATION_AUDIO_SIGNAL_CLASS, 1984
AGS_IS_FX_NOTATION_CHANNEL, 1986
AGS_IS_FX_NOTATION_CHANNEL_CLASS, 1986
AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR, 1989
AGS_IS_FX_NOTATION_CHANNEL_PROCESSOR_CLASS, 1989
AGS_IS_FX_NOTATION_RECYCLING, 1991
AGS_IS_FX_NOTATION_RECYCLING_CLASS, 1991
AGS_IS_FX_PATTERN_AUDIO, 1954
AGS_IS_FX_PATTERN_AUDIO_CLASS, 1954
AGS_IS_FX_PATTERN_AUDIO_PROCESSOR, 1959
AGS_IS_FX_PATTERN_AUDIO_PROCESSOR_CLASS, 1959
AGS_IS_FX_PATTERN_AUDIO_SIGNAL, 1962
AGS_IS_FX_PATTERN_AUDIO_SIGNAL_CLASS, 1962
AGS_IS_FX_PATTERN_CHANNEL, 1964
AGS_IS_FX_PATTERN_CHANNEL_CLASS, 1964
AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR, 1967
AGS_IS_FX_PATTERN_CHANNEL_PROCESSOR_CLASS, 1967
AGS_IS_FX_PATTERN_RECYCLING, 1969
AGS_IS_FX_PATTERN_RECYCLING_CLASS, 1969
AGS_IS_FX_PEAK_AUDIO, 2109
AGS_IS_FX_PEAK_AUDIO_CLASS, 2109
AGS_IS_FX_PEAK_AUDIO_PROCESSOR, 2111
AGS_IS_FX_PEAK_AUDIO_PROCESSOR_CLASS, 2111
AGS_IS_FX_PEAK_AUDIO_SIGNAL, 2113
AGS_IS_FX_PEAK_AUDIO_SIGNAL_CLASS, 2113
AGS_IS_FX_PEAK_CHANNEL, 2116
AGS_IS_FX_PEAK_CHANNEL_CLASS, 2117
AGS_IS_FX_PEAK_CHANNEL_PROCESSOR, 2119
AGS_IS_FX_PEAK_CHANNEL_PROCESSOR_CLASS, 2119
AGS_IS_FX_PEAK_RECYCLING, 2121
AGS_IS_FX_PEAK_RECYCLING_CLASS, 2121
AGS_IS_FX_PLAYBACK_AUDIO, 1916
AGS_IS_FX_PLAYBACK_AUDIO_CLASS, 1916
AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR, 1924
AGS_IS_FX_PLAYBACK_AUDIO_PROCESSOR_CLASS, 1924
AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL, 1927
AGS_IS_FX_PLAYBACK_AUDIO_SIGNAL_CLASS, 1927
AGS_IS_FX_PLAYBACK_CHANNEL, 1929
AGS_IS_FX_PLAYBACK_CHANNEL_CLASS, 1929
AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR, 1932
AGS_IS_FX_PLAYBACK_CHANNEL_PROCESSOR_CLASS, 1932
AGS_IS_FX_PLAYBACK_RECYCLING, 1934
AGS_IS_FX_PLAYBACK_RECYCLING_CLASS, 1934
AGS_IS_FX_SF2_SYNTH_AUDIO, 2091
AGS_IS_FX_SF2_SYNTH_AUDIO_CLASS, 2091
AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR, 2096
AGS_IS_FX_SF2_SYNTH_AUDIO_PROCESSOR_CLASS, 2096
AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL, 2099
AGS_IS_FX_SF2_SYNTH_AUDIO_SIGNAL_CLASS, 2099
AGS_IS_FX_SF2_SYNTH_CHANNEL, 2101
AGS_IS_FX_SF2_SYNTH_CHANNEL_CLASS, 2101
AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR, 2103
AGS_IS_FX_SF2_SYNTH_CHANNEL_PROCESSOR_CLASS, 2103
AGS_IS_FX_SF2_SYNTH_RECYCLING, 2106
AGS_IS_FX_SF2_SYNTH_RECYCLING_CLASS, 2106
AGS_IS_FX_SYNTH_AUDIO, 2036
AGS_IS_FX_SYNTH_AUDIO_CLASS, 2037
AGS_IS_FX_SYNTH_AUDIO_PROCESSOR, 2050
AGS_IS_FX_SYNTH_AUDIO_PROCESSOR_CLASS, 2050
AGS_IS_FX_SYNTH_AUDIO_SIGNAL, 2052
AGS_IS_FX_SYNTH_AUDIO_SIGNAL_CLASS, 2052
AGS_IS_FX_SYNTH_CHANNEL, 2054

AGS_IS_FX_SYNTH_CHANNEL_CLASS, 2055
AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR, 2057
AGS_IS_FX_SYNTH_CHANNEL_PROCESSOR_CLASS, 2057
AGS_IS_FX_SYNTH_RECYCLING, 2059
AGS_IS_FX_SYNTH_RECYCLING_CLASS, 2059
AGS_IS_FX_TWO_PASS_ALIASE_AUDIO, 2251
AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_CLASS, 2251
AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR, 2254
AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR_CLASS, 2254
AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL, 2256
AGS_IS_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL_CLASS, 2256
AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL, 2263
AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_CLASS, 2263
AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR, 2266
AGS_IS_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR_CLASS, 2266
AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING, 2268
AGS_IS_FX_TWO_PASS_ALIASE_RECYCLING_CLASS, 2268
AGS_IS_FX_VOLUME_AUDIO, 2142
AGS_IS_FX_VOLUME_AUDIO_CLASS, 2142
AGS_IS_FX_VOLUME_AUDIO_PROCESSOR, 2144
AGS_IS_FX_VOLUME_AUDIO_PROCESSOR_CLASS, 2144
AGS_IS_FX_VOLUME_AUDIO_SIGNAL, 2146
AGS_IS_FX_VOLUME_AUDIO_SIGNAL_CLASS, 2146
AGS_IS_FX_VOLUME_CHANNEL, 2149
AGS_IS_FX_VOLUME_CHANNEL_CLASS, 2149
AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR, 2152
AGS_IS_FX_VOLUME_CHANNEL_PROCESSOR_CLASS, 2152
AGS_IS_FX_VOLUME_RECYCLING, 2154
AGS_IS_FX_VOLUME_RECYCLING_CLASS, 2154
AGS_IS_GENERIC_RECALL_CHANNEL_RUN, 1445
AGS_IS_GENERIC_RECALL_CHANNEL_RUN_CLASS, 1445
AGS_IS_GENERIC_RECALL_RECYCLING, 1448
AGS_IS_GENERIC_RECALL_RECYCLING_CLASS, 1448
AGS_IS_GSTREAMER_CLIENT, 826
AGS_IS_GSTREAMER_CLIENT_CLASS, 826
AGS_IS_GSTREAMER_DEVIN, 840
AGS_IS_GSTREAMER_DEVIN_CLASS, 840
AGS_IS_GSTREAMER_DEVOUT, 850
AGS_IS_GSTREAMER_DEVOUT_CLASS, 850
AGS_IS_GSTREAMER_FILE, 1824
AGS_IS_GSTREAMER_FILE_CLASS, 1825
AGS_IS_GSTREAMER_PORT, 833
AGS_IS_GSTREAMER_PORT_CLASS, 833
AGS_IS_GSTREAMER_SERVER, 818
AGS_IS_GSTREAMER_SERVER_CLASS, 818
AGS_IS_INPUT, 1103
AGS_IS_INPUT_CLASS, 1103
AGS_IS_IPATCH, 1830
AGS_IS_IPATCH_CLASS, 1830
AGS_IS_IPATCH_DLS2_READER, 1852
AGS_IS_IPATCH_DLS2_READER_CLASS, 1852
AGS_IS_IPATCH_GIG_READER, 1837
AGS_IS_IPATCH_GIG_READER_CLASS, 1837
AGS_IS_IPATCH_SAMPLE, 1855
AGS_IS_IPATCH_SAMPLE_CLASS, 1856
AGS_IS_IPATCH_SF2_READER, 1845
AGS_IS_IPATCH_SF2_READER_CLASS, 1845
AGS_IS_JACK_CLIENT, 733
AGS_IS_JACK_CLIENT_CLASS, 733
AGS_IS_JACK_DEVIN, 745
AGS_IS_JACK_DEVIN_CLASS, 745
AGS_IS_JACK_DEVOUT, 755
AGS_IS_JACK_DEVOUT_CLASS, 755
AGS_IS_JACK_MIDIIN, 765
AGS_IS_JACK_MIDIIN_CLASS, 765
AGS_IS_JACK_PORT, 739
AGS_IS_JACK_PORT_CLASS, 739
AGS_IS_JACK_SERVER, 725
AGS_IS_JACK_SERVER_CLASS, 725
AGS_IS_LADSPA_CONVERSION, 41
AGS_IS_LADSPA_CONVERSION_CLASS, 41
AGS_IS_LADSPA_MANAGER, 49
AGS_IS_LADSPA_MANAGER_CLASS, 49
AGS_IS_LADSPA_PLUGIN, 51
AGS_IS_LADSPA_PLUGIN_CLASS, 51
AGS_IS_LINK_CHANNEL, 2338
AGS_IS_LINK_CHANNEL_CLASS, 2338
AGS_IS_LV2_CONVERSION, 65
AGS_IS_LV2_CONVERSION_CLASS, 65
AGS_IS_LV2_EVENT_MANAGER, 69
AGS_IS_LV2_EVENT_MANAGER_CLASS, 69
AGS_IS_LV2_LOG_MANAGER, 72
AGS_IS_LV2_LOG_MANAGER_CLASS, 72
AGS_IS_LV2_MANAGER, 80
AGS_IS_LV2_MANAGER_CLASS, 80
AGS_IS_LV2_OPTION_MANAGER, 86
AGS_IS_LV2_OPTION_MANAGER_CLASS, 86
AGS_IS_LV2_PLUGIN, 107
AGS_IS_LV2_PLUGIN_CLASS, 107
AGS_IS_LV2_PRESET, 117
AGS_IS_LV2_PRESET_CLASS, 117
AGS_IS_LV2_PRESET_MANAGER, 112
AGS_IS_LV2_PRESET_MANAGER_CLASS, 112
AGS_IS_LV2_TURTLE_PARSER, 91
AGS_IS_LV2_TURTLE_PARSER_CLASS, 91
AGS_IS_LV2_TURTLE_SCANNER, 96
AGS_IS_LV2_TURTLE_SCANNER_CLASS, 96
AGS_IS_LV2_URI_MAP_MANAGER, 123
AGS_IS_LV2_URI_MAP_MANAGER_CLASS, 123
AGS_IS_LV2_URID_MANAGER, 127
AGS_IS_LV2_URID_MANAGER_CLASS, 127
AGS_IS_LV2_WORKER, 132
AGS_IS_LV2_WORKER_CLASS, 132
AGS_IS_LV2_WORKER_MANAGER, 137

AGS_IS_LV2_WORKER_MANAGER_CLASS, 137
AGS_IS_LV2UI_MANAGER, 142
AGS_IS_LV2UI_MANAGER_CLASS, 142
AGS_IS_LV2UI_PLUGIN, 146
AGS_IS_LV2UI_PLUGIN_CLASS, 146
AGS_IS_MIDI, 1281
AGS_IS_MIDI_BUILDER, 1594
AGS_IS_MIDI_BUILDER_CLASS, 1594
AGS_IS_MIDI_CLASS, 1281
AGS_IS_MIDI_FILE, 1563
AGS_IS_MIDI_FILE_CLASS, 1563
AGS_IS_MIDI_PARSER, 1618
AGS_IS_MIDI_PARSER_CLASS, 1619
AGS_IS_MOVE_NOTE, 2341
AGS_IS_MOVE_NOTE_CLASS, 2342
AGS_IS_NOTATION, 1181
AGS_IS_NOTATION_CLASS, 1181
AGS_IS_NOTE, 1203
AGS_IS_NOTE_CLASS, 1203
AGS_IS_OPEN_FILE, 2346
AGS_IS_OPEN_FILE_CLASS, 2346
AGS_IS_OPEN_SF2_INSTRUMENT, 2353
AGS_IS_OPEN_SF2_INSTRUMENT_CLASS, 2353
AGS_IS_OPEN_SF2_SAMPLE, 2357
AGS_IS_OPEN_SF2_SAMPLE_CLASS, 2357
AGS_IS_OPEN_SFZ_FILE, 2361
AGS_IS_OPEN_SFZ_FILE_CLASS, 2361
AGS_IS_OPEN_SINGLE_FILE, 2350
AGS_IS_OPEN_SINGLE_FILE_CLASS, 2350
AGS_IS_OPEN_WAVE, 2364
AGS_IS_OPEN_WAVE_CLASS, 2364
AGS_IS_OSC_ACTION_CONTROLLER, 1720
AGS_IS_OSC_ACTION_CONTROLLER_CLASS, 1720
AGS_IS_OSC_BUILDER, 1659
AGS_IS_OSC_BUILDER_CLASS, 1659
AGS_IS_OSC_CLIENT, 1681
AGS_IS_OSC_CLIENT_CLASS, 1681
AGS_IS_OSC_CONFIG_CONTROLLER, 1724
AGS_IS_OSC_CONFIG_CONTROLLER_CLASS, 1724
AGS_IS_OSC_CONNECTION, 1703
AGS_IS_OSC_CONNECTION_CLASS, 1703
AGS_IS_OSC_CONTROLLER, 1717
AGS_IS_OSC_CONTROLLER_CLASS, 1717
AGS_IS_OSC_EXPORT_CONTROLLER, 1727
AGS_IS_OSC_EXPORT_CONTROLLER_CLASS, 1727
AGS_IS_OSC_FRONT_CONTROLLER, 1732
AGS_IS_OSC_FRONT_CONTROLLER_CLASS, 1732
AGS_IS_OSC_INFO_CONTROLLER, 1737
AGS_IS_OSC_INFO_CONTROLLER_CLASS, 1737
AGS_IS_OSC_MESSAGE, 1708
AGS_IS_OSC_MESSAGE_CLASS, 1708
AGS_IS_OSC_METER_CONTROLLER, 1744
AGS_IS_OSC_METER_CONTROLLER_CLASS, 1744
AGS_IS_OSC_NODE_CONTROLLER, 1749
AGS_IS_OSC_NODE_CONTROLLER_CLASS, 1749
AGS_IS_OSC_PARSER, 1671
AGS_IS_OSC_PARSER_CLASS, 1671
AGS_IS_OSC_PLUGIN_CONTROLLER, 1752
AGS_IS_OSC_PLUGIN_CONTROLLER_INTERFACE, 1752
AGS_IS_OSC_RENEW_CONTROLLER, 1754
AGS_IS_OSC_RENEW_CONTROLLER_CLASS, 1755
AGS_IS_OSC_RESPONSE, 1713
AGS_IS_OSC_RESPONSE_CLASS, 1713
AGS_IS_OSC_SERVER, 1692
AGS_IS_OSC_SERVER_CLASS, 1692
AGS_IS_OSC_STATUS_CONTROLLER, 1758
AGS_IS_OSC_STATUS_CONTROLLER_CLASS, 1758
AGS_IS_OSC_WEBSOCKET_CONNECTION, 1765
AGS_IS_OSC_WEBSOCKET_CONNECTION_CLASS, 1765
AGS_IS_OSC_XMLRPC_CONTROLLER, 1774
AGS_IS_OSC_XMLRPC_CONTROLLER_CLASS, 1774
AGS_IS_OSC_XMLRPC_MESSAGE, 1768
AGS_IS_OSC_XMLRPC_MESSAGE_CLASS, 1768
AGS_IS_OSC_XMLRPC_SERVER, 1762
AGS_IS_OSC_XMLRPC_SERVER_CLASS, 1762
AGS_IS_OSS_DEVIN, 677
AGS_IS_OSS_DEVIN_CLASS, 677
AGS_IS_OSS_DEVOUT, 687
AGS_IS_OSS_DEVOUT_CLASS, 687
AGS_IS_OSS_MIDIIN, 696
AGS_IS_OSS_MIDIIN_CLASS, 696
AGS_IS_OUTPUT, 1098
AGS_IS_OUTPUT_CLASS, 1098
AGS_IS_PATTERN, 1299
AGS_IS_PATTERN_CLASS, 1299
AGS_IS_PLAYBACK, 1333
AGS_IS_PLAYBACK_CLASS, 1333
AGS_IS_PLAYBACK_DOMAIN, 1326
AGS_IS_PLAYBACK_DOMAIN_CLASS, 1326
AGS_IS_PLUGIN_PORT, 34
AGS_IS_PLUGIN_PORT_CLASS, 35
AGS_IS_PORT, 1156
AGS_IS_PORT_CLASS, 1156
AGS_IS_PRESET, 1314
AGS_IS_PRESET_CLASS, 1314
AGS_IS_PULSE_CLIENT, 782
AGS_IS_PULSE_CLIENT_CLASS, 782
AGS_IS_PULSE_DEVIN, 796
AGS_IS_PULSE_DEVIN_CLASS, 796
AGS_IS_PULSE_DEVOUT, 806
AGS_IS_PULSE_DEVOUT_CLASS, 806
AGS_IS_PULSE_PORT, 790
AGS_IS_PULSE_PORT_CLASS, 790
AGS_IS_PULSE_SERVER, 775
AGS_IS_PULSE_SERVER_CLASS, 775
AGS_IS_RECALL, 1401
AGS_IS_RECALL_AUDIO, 1418
AGS_IS_RECALL_AUDIO_CLASS, 1419
AGS_IS_RECALL_AUDIO_RUN, 1423
AGS_IS_RECALL_AUDIO_RUN_CLASS, 1423
AGS_IS_RECALL_AUDIO_SIGNAL, 1442
AGS_IS_RECALL_AUDIO_SIGNAL_CLASS, 1442
AGS_IS_RECALL_CHANNEL, 1427
AGS_IS_RECALL_CHANNEL_CLASS, 1428

AGS_IS_RECALL_CHANNEL_RUN, 1434
AGS_IS_RECALL_CHANNEL_RUN_CLASS, 1434
AGS_IS_RECALL_CLASS, 1401
AGS_IS_RECALL_CONTAINER, 1356
AGS_IS_RECALL_CONTAINER_CLASS, 1357
AGS_IS_RECALL_DEPENDENCY, 1363
AGS_IS_RECALL_DEPENDENCY_CLASS, 1363
AGS_IS_RECALL_ID, 1341
AGS_IS_RECALL_ID_CLASS, 1341
AGS_IS_RECALL_RECYCLING, 1438
AGS_IS_RECALL_RECYCLING_CLASS, 1438
AGS_IS_RECYCLING, 1116
AGS_IS_RECYCLING_CLASS, 1116
AGS_IS_RECYCLING_CONTEXT, 1349
AGS_IS_RECYCLING_CONTEXT_CLASS, 1349
AGS_IS_REMOVE_AUDIO, 2367
AGS_IS_REMOVE_AUDIO_CLASS, 2367
AGS_IS_REMOVE_AUDIO_SIGNAL, 2370
AGS_IS_REMOVE_AUDIO_SIGNAL_CLASS, 2370
AGS_IS_REMOVE_NOTE, 2373
AGS_IS_REMOVE_NOTE_CLASS, 2373
AGS_IS_REMOVE_SOUNDCARD, 2376
AGS_IS_REMOVE_SOUNDCARD_CLASS, 2376
AGS_IS_RESET_FX_ANALYSE, 2385
AGS_IS_RESET_FX_ANALYSE_CLASS, 2385
AGS_IS_RESET_FX_PEAK, 2382
AGS_IS_RESET_FX_PEAK_CLASS, 2382
AGS_IS_RESET_NOTE, 2379
AGS_IS_RESET_NOTE_CLASS, 2379
AGS_IS_RESIZE_AUDIO, 2388
AGS_IS_RESIZE_AUDIO_CLASS, 2388
AGS_IS_SEEK_SOUNDCARD, 2391
AGS_IS_SEEK_SOUNDCARD_CLASS, 2391
AGS_IS_SEQUENCER_THREAD, 1479
AGS_IS_SEQUENCER_THREAD_CLASS, 1480
AGS_IS_SET_AUDIO_CHANNELS, 2394
AGS_IS_SET_AUDIO_CHANNELS_CLASS, 2395
AGS_IS_SET_BUFFER_SIZE, 2397
AGS_IS_SET_BUFFER_SIZE_CLASS, 2397
AGS_IS_SET_DEVICE, 2400
AGS_IS_SET_DEVICE_CLASS, 2400
AGS_IS_SET_FORMAT, 2403
AGS_IS_SET_FORMAT_CLASS, 2403
AGS_IS_SET_MUTED, 2406
AGS_IS_SET_MUTED_CLASS, 2406
AGS_IS_SET_SAMPLERATE, 2409
AGS_IS_SET_SAMPLERATE_CLASS, 2409
AGS_IS_SF2_LOADER, 1491
AGS_IS_SF2_LOADER_CLASS, 1491
AGS_IS_SF2_MIDI_LOCALE_LOADER, 1501
AGS_IS_SF2_MIDI_LOCALE_LOADER_CLASS, 1501
AGS_IS_SF2_SYNTH_GENERATOR, 529
AGS_IS_SF2_SYNTH_GENERATOR_CLASS, 529
AGS_IS_SFZ_FILE, 1868
AGS_IS_SFZ_FILE_CLASS, 1868
AGS_IS_SFZ_GROUP, 1876
AGS_IS_SFZ_GROUP_CLASS, 1876
AGS_IS_SFZ_INSTRUMENT_LOADER, 1518
AGS_IS_SFZ_INSTRUMENT_LOADER_CLASS, 1518
AGS_IS_SFZ_LOADER, 1509
AGS_IS_SFZ_LOADER_CLASS, 1509
AGS_IS_SFZ_REGION, 1882
AGS_IS_SFZ_REGION_CLASS, 1882
AGS_IS_SFZ_SAMPLE, 1890
AGS_IS_SFZ_SAMPLE_CLASS, 1891
AGS_IS_SFZ_SYNTH_GENERATOR, 546
AGS_IS_SFZ_SYNTH_GENERATOR_CLASS, 546
AGS_IS_SNDFILE, 1860
AGS_IS_SNDFILE_CLASS, 1860
AGS_IS_SOUND_CONTAINER, 1900
AGS_IS_SOUND_CONTAINER_INTERFACE, 1901
AGS_IS_SOUND_PROVIDER, 954
AGS_IS_SOUND_PROVIDER_INTERFACE, 954
AGS_IS_SOUND_RESOURCE, 1908
AGS_IS_SOUND_RESOURCE_INTERFACE, 1908
AGS_IS_SOUNDCARD_THREAD, 1483
AGS_IS_SOUNDCARD_THREAD_CLASS, 1483
AGS_IS_START_AUDIO, 2412
AGS_IS_START_AUDIO_CLASS, 2412
AGS_IS_START_CHANNEL, 2415
AGS_IS_START_CHANNEL_CLASS, 2415
AGS_IS_START_SEQUENCER, 2417
AGS_IS_START_SEQUENCER_CLASS, 2418
AGS_IS_START_SOUNDCARD, 2420
AGS_IS_START_SOUNDCARD_CLASS, 2420
AGS_IS_STOP_SEQUENCER, 2422
AGS_IS_STOP_SEQUENCER_CLASS, 2422
AGS_IS_STOP_SOUNDCARD, 2424
AGS_IS_STOP_SOUNDCARD_CLASS, 2424
AGS_IS_STOP_THREAD, 2426
AGS_IS_STOP_THREAD_CLASS, 2426
AGS_IS_SWITCH_BUFFER_FLAG, 2428
AGS_IS_SWITCH_BUFFER_FLAG_CLASS, 2428
AGS_IS_SYNTH_GENERATOR, 506
AGS_IS_SYNTH_GENERATOR_CLASS, 506
AGS_IS_TIC_DEVICE, 2431
AGS_IS_TIC_DEVICE_CLASS, 2431
AGS_IS_TOGGLE_PATTERN_BIT, 2434
AGS_IS_TOGGLE_PATTERN_BIT_CLASS, 2434
AGS_IS_TRACK, 1290
AGS_IS_TRACK_CLASS, 1290
AGS_IS_WASAPI_DEVIN, 714
AGS_IS_WASAPI_DEVIN_CLASS, 714
AGS_IS_WASAPI_DEVOUT, 705
AGS_IS_WASAPI_DEVOUT_CLASS, 705
AGS_IS_WAVE, 1260
AGS_IS_WAVE_CLASS, 1260
AGS_IS_WAVE_LOADER, 1525
AGS_IS_WAVE_LOADER_CLASS, 1525

J

AGS_JACK_CLIENT, 733
ags_jack_client_activate, 732
ags_jack_client_add_device, 731

ags_jack_client_add_port, 732
AGS_JACK_CLIENT_CLASS, 733
ags_jack_client_close, 731
ags_jack_client_deactivate, 733
ags_jack_client_find, 730
ags_jack_client_find_uuid, 730
ags_jack_client_flags_get_type, 734
AGS_JACK_CLIENT_GET_CLASS, 734
AGS_JACK_CLIENT_GET_OBJ_MUTEX, 728
ags_jack_client_get_type, 734
ags_jack_client_new, 733
ags_jack_client_open, 730
ags_jack_client_remove_device, 731
ags_jack_client_remove_port, 732
ags_jack_client_set_flags, 729
ags_jack_client_test_flags, 729
ags_jack_client_unset_flags, 729
AGS_JACK_DEVIN, 745
ags_jack_devin_adjust_delay_and_attack, 744
AGS_JACK_DEVIN_CLASS, 746
AGS_JACK_DEVIN_ERROR, 748
ags_jack_devin_error_quark, 743
ags_jack_devin_flags_get_type, 746
AGS_JACK_DEVIN_GET_CLASS, 746
AGS_JACK_DEVIN_GET_OBJ_MUTEX, 743
ags_jack_devin_get_type, 746
ags_jack_devin_new, 745
ags_jack_devin_realloc_buffer, 745
ags_jack_devin_set_flags, 744
ags_jack_devin_switch_buffer_flag, 744
ags_jack_devin_test_flags, 743
ags_jack_devin_unset_flags, 744
AGS_JACK_DEVOUT, 755
ags_jack_devout_adjust_delay_and_attack, 754
AGS_JACK_DEVOUT_CLASS, 756
AGS_JACK_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 756
AGS_JACK_DEVOUT_ERROR, 758
ags_jack_devout_error_quark, 753
ags_jack_devout_flags_get_type, 756
AGS_JACK_DEVOUT_GET_CLASS, 756
AGS_JACK_DEVOUT_GET_OBJ_MUTEX, 753
ags_jack_devout_get_type, 756
ags_jack_devout_new, 755
ags_jack_devout_realloc_buffer, 755
ags_jack_devout_set_flags, 754
ags_jack_devout_switch_buffer_flag, 754
ags_jack_devout_test_flags, 753
ags_jack_devout_unset_flags, 754
AGS_JACK_MIDIIN, 765
AGS_JACK_MIDIIN_CLASS, 765
AGS_JACK_MIDIIN_DEFAULT_BUFFER_SIZE, 765
AGS_JACK_MIDIIN_ERROR, 767
ags_jack_midiin_error_quark, 763
ags_jack_midiin_flags_get_type, 765
AGS_JACK_MIDIIN_GET_CLASS, 765
AGS_JACK_MIDIIN_GET_OBJ_MUTEX, 763
ags_jack_midiin_get_type, 765
ags_jack_midiin_new, 764
ags_jack_midiin_set_flags, 763
ags_jack_midiin_switch_buffer_flag, 764
ags_jack_midiin_test_flags, 763
ags_jack_midiin_unset_flags, 764
AGS_JACK_PORT, 739
AGS_JACK_PORT_CLASS, 739
ags_jack_port_find, 738
ags_jack_port_flags_get_type, 740
AGS_JACK_PORT_GET_CLASS, 740
AGS_JACK_PORT_GET_OBJ_MUTEX, 737
ags_jack_port_get_type, 740
ags_jack_port_new, 739
ags_jack_port_register, 738
ags_jack_port_set_flags, 737
ags_jack_port_test_flags, 737
ags_jack_port_unregister, 739
ags_jack_port_unset_flags, 737
AGS_JACK_SERVER, 725
ags_jack_server_add_client, 723
AGS_JACK_SERVER_CLASS, 725
ags_jack_server_connect_client, 724
AGS_JACK_SERVER_DEFAULT_HOST, 725
AGS_JACK_SERVER_DEFAULT_PORT, 725
AGS_JACK_SERVER_DEFAULT_PROTOCOL, 725
ags_jack_server_disconnect_client, 724
ags_jack_server_find_client, 722
ags_jack_server_find_port, 723
ags_jack_server_find_url, 722
AGS_JACK_SERVER_GET_CLASS, 725
AGS_JACK_SERVER_GET_OBJ_MUTEX, 721
ags_jack_server_get_type, 725
ags_jack_server_new, 724
ags_jack_server_remove_client, 723
ags_jack_server_set_flags, 721
ags_jack_server_test_flags, 721
ags_jack_server_unset_flags, 722
AgsJackClient, 734
AgsJackClient:client-name, 735
AgsJackClient:device, 735
AgsJackClient:jack-server, 735
AgsJackClient:port, 735
AgsJackClientClass, 734
AgsJackClientFlags, 734
AgsJackDevin, 748
AgsJackDevin:attack, 749
AgsJackDevin:bpm, 749
AgsJackDevin:buffer, 749
AgsJackDevin:buffer-size, 749
AgsJackDevin:delay-factor, 750
AgsJackDevin:device, 750
AgsJackDevin:dsp-channels, 750
AgsJackDevin:format, 750
AgsJackDevin:jack-client, 751
AgsJackDevin:jack-port, 751
AgsJackDevin:pcm-channels, 751
AgsJackDevin:samplerate, 751

AgsJackDevinAppBufferMode, 747
AgsJackDevinClass, 748
AgsJackDevinError, 748
AgsJackDevinFlags, 746
AgsJackDevinSyncFlags, 747
AgsJackDevout, 758
AgsJackDevout:attack, 759
AgsJackDevout:bpm, 759
AgsJackDevout:buffer, 759
AgsJackDevout:buffer-size, 759
AgsJackDevout:delay-factor, 760
AgsJackDevout:device, 760
AgsJackDevout:dsp-channels, 760
AgsJackDevout:format, 760
AgsJackDevout:jack-client, 761
AgsJackDevout:jack-port, 761
AgsJackDevout:pcm-channels, 761
AgsJackDevout:samplerate, 761
AgsJackDevoutAppBufferMode, 757
AgsJackDevoutClass, 759
AgsJackDevoutError, 758
AgsJackDevoutFlags, 756
AgsJackDevoutSyncFlags, 757
AgsJackMidiin, 768
AgsJackMidiin:attack, 768
AgsJackMidiin:bpm, 768
AgsJackMidiin:buffer, 768
AgsJackMidiin:delay-factor, 769
AgsJackMidiin:device, 769
AgsJackMidiin:jack-client, 769
AgsJackMidiin:jack-port, 769
AgsJackMidiinAppBufferMode, 766
AgsJackMidiinClass, 768
AgsJackMidiinError, 767
AgsJackMidiinFlags, 765
AgsJackMidiinSyncFlags, 766
AgsJackPort, 741
AgsJackPort:jack-client, 741
AgsJackPort:port-name, 741
AgsJackPortClass, 741
AgsJackPortFlags, 740
AgsJackServer, 726
AgsJackServer:default-jack-client, 726
AgsJackServer:default-soundcard, 726
AgsJackServer:input-jack-client, 726
AgsJackServer:jack-client, 727
AgsJackServer:url, 727
AgsJackServerClass, 726

L

AGS_LADSPA_CONVERSION, 41
AGS_LADSPA_CONVERSION_CLASS, 42
AGS_LADSPA_CONVERSION_DEFAULT_LOWER, 42
AGS_LADSPA_CONVERSION_DEFAULT_STEP_COUNT, 42
AGS_LADSPA_CONVERSION_DEFAULT_UPPER, 42
ags_ladspa_conversion_flags_get_type, 42
AGS_LADSPA_CONVERSION_GET_CLASS, 42
ags_ladspa_conversion_get_type, 42
ags_ladspa_conversion_new, 41
ags_ladspa_conversion_set_flags, 40
ags_ladspa_conversion_test_flags, 40
ags_ladspa_conversion_unset_flags, 41
AGS_LADSPA_MANAGER, 49
AGS_LADSPA_MANAGER_CLASS, 49
ags_ladspa_manager_find_ladspa_plugin, 46
ags_ladspa_manager_find_ladspa_plugin_with_fallback, 47
AGS_LADSPA_MANAGER_GET_CLASS, 49
ags_ladspa_manager_get_default_path, 45
ags_ladspa_manager_get_filenames, 46
ags_ladspa_manager_get_instance, 48
ags_ladspa_manager_get_ladspa_plugin, 48
AGS_LADSPA_MANAGER_GET_OBJ_MUTEX, 45
ags_ladspa_manager_get_type, 49
ags_ladspa_manager_load_blacklist, 47
ags_ladspa_manager_load_default_directory, 48
ags_ladspa_manager_load_file, 47
ags_ladspa_manager_new, 49
ags_ladspa_manager_set_default_path, 45
AGS_LADSPA_PLUGIN, 51
AGS_LADSPA_PLUGIN_CLASS, 52
AGS_LADSPA_PLUGIN_DESCRIPTOR, 51
AGS_LADSPA_PLUGIN_GET_CLASS, 52
ags_ladspa_plugin_get_type, 52
ags_ladspa_plugin_new, 51
AgsLadspaConversion, 43
AgsLadspaConversion:lower, 43
AgsLadspaConversion:samplerate, 43
AgsLadspaConversion:step-count, 44
AgsLadspaConversion:upper, 44
AgsLadspaConversionClass, 43
AgsLadspaConversionFlags, 42
AgsLadspaManager, 50
AgsLadspaManagerClass, 50
AgsLadspaPlugin, 52
AgsLadspaPlugin:unique-id, 52
AgsLadspaPluginClass, 52
ags_lfo_synth_util_alloc, 468
ags_lfo_synth_util_compute_impulse, 489
ags_lfo_synth_util_compute_impulse_complex, 489
ags_lfo_synth_util_compute_impulse_double, 488
ags_lfo_synth_util_compute_impulse_float, 488
ags_lfo_synth_util_compute_impulse_s16, 487
ags_lfo_synth_util_compute_impulse_s24, 487
ags_lfo_synth_util_compute_impulse_s32, 487
ags_lfo_synth_util_compute_impulse_s64, 488
ags_lfo_synth_util_compute_impulse_s8, 487
ags_lfo_synth_util_compute_sawtooth, 482
ags_lfo_synth_util_compute_sawtooth_complex, 481
ags_lfo_synth_util_compute_sawtooth_double, 481
ags_lfo_synth_util_compute_sawtooth_float, 481
ags_lfo_synth_util_compute_sawtooth_s16, 480
ags_lfo_synth_util_compute_sawtooth_s24, 480
ags_lfo_synth_util_compute_sawtooth_s32, 480

ags_lfo_synth_util_compute_sawtooth_s64, 480
ags_lfo_synth_util_compute_sawtooth_s8, 479
ags_lfo_synth_util_compute_sin, 479
ags_lfo_synth_util_compute_sin_complex, 479
ags_lfo_synth_util_compute_sin_double, 478
ags_lfo_synth_util_compute_sin_float, 478
ags_lfo_synth_util_compute_sin_s16, 477
ags_lfo_synth_util_compute_sin_s24, 477
ags_lfo_synth_util_compute_sin_s32, 478
ags_lfo_synth_util_compute_sin_s64, 478
ags_lfo_synth_util_compute_sin_s8, 477
ags_lfo_synth_util_compute_square, 486
ags_lfo_synth_util_compute_square_complex, 486
ags_lfo_synth_util_compute_square_double, 486
ags_lfo_synth_util_compute_square_float, 485
ags_lfo_synth_util_compute_square_s16, 484
ags_lfo_synth_util_compute_square_s24, 485
ags_lfo_synth_util_compute_square_s32, 485
ags_lfo_synth_util_compute_square_s64, 485
ags_lfo_synth_util_compute_square_s8, 484
ags_lfo_synth_util_compute_triangle, 484
ags_lfo_synth_util_compute_triangle_complex, 484
ags_lfo_synth_util_compute_triangle_double, 484
ags_lfo_synth_util_compute_triangle_float, 483
ags_lfo_synth_util_compute_triangle_s16, 482
ags_lfo_synth_util_compute_triangle_s24, 482
ags_lfo_synth_util_compute_triangle_s32, 483
ags_lfo_synth_util_compute_triangle_s64, 483
ags_lfo_synth_util_compute_triangle_s8, 482
ags_lfo_synth_util_copy, 469
AGS_LFO_SYNTH_UTIL_DEFAULT_FREQUENCY, 489
AGS_LFO_SYNTH_UTIL_DEFAULT_LFO_DEPTH, 489
AGS_LFO_SYNTH_UTIL_DEFAULT_TUNING, 489
ags_lfo_synth_util_free, 469
ags_lfo_synth_util_get_buffer_length, 470
ags_lfo_synth_util_get_format, 471
ags_lfo_synth_util_get_frame_count, 476
ags_lfo_synth_util_get_frequency, 473
ags_lfo_synth_util_get_lfo_depth, 474
ags_lfo_synth_util_get_lfo_synth_oscillator_mode, 472
ags_lfo_synth_util_get_offset, 476
ags_lfo_synth_util_get_phase, 474
ags_lfo_synth_util_get_samplerate, 472
ags_lfo_synth_util_get_source, 469
ags_lfo_synth_util_get_source_stride, 470
ags_lfo_synth_util_get_tuning, 475
ags_lfo_synth_util_get_type, 489
ags_lfo_synth_util_set_buffer_length, 471
ags_lfo_synth_util_set_format, 471
ags_lfo_synth_util_set_frame_count, 477
ags_lfo_synth_util_set_frequency, 473
ags_lfo_synth_util_set_lfo_depth, 475
ags_lfo_synth_util_set_lfo_synth_oscillator_mode, 473
ags_lfo_synth_util_set_offset, 476
ags_lfo_synth_util_set_phase, 474
ags_lfo_synth_util_set_samplerate, 472
ags_lfo_synth_util_set_source, 470
ags_lfo_synth_util_set_source_stride, 470
ags_lfo_synth_util_set_tuning, 475
AgsLFOSynthUtil, 490
AGS_LINEAR_INTERPOLATE_UTIL, 200
ags_linear_interpolate_util_alloc, 191
ags_linear_interpolate_util_copy, 192
ags_linear_interpolate_util_free, 192
ags_linear_interpolate_util_get_buffer_length, 195
ags_linear_interpolate_util_get_destination, 194
ags_linear_interpolate_util_get_destination_stride, 194
ags_linear_interpolate_util_get_factor, 197
ags_linear_interpolate_util_get_format, 196
ags_linear_interpolate_util_get_samplerate, 196
ags_linear_interpolate_util_get_source, 192
ags_linear_interpolate_util_get_source_stride, 193
ags_linear_interpolate_util_get_type, 200
ags_linear_interpolate_util_pitch, 200
ags_linear_interpolate_util_pitch_complex, 200
ags_linear_interpolate_util_pitch_double, 199
ags_linear_interpolate_util_pitch_float, 199
ags_linear_interpolate_util_pitch_s16, 198
ags_linear_interpolate_util_pitch_s24, 198
ags_linear_interpolate_util_pitch_s32, 199
ags_linear_interpolate_util_pitch_s64, 199
ags_linear_interpolate_util_pitch_s8, 198
ags_linear_interpolate_util_set_buffer_length, 195
ags_linear_interpolate_util_set_destination, 194
ags_linear_interpolate_util_set_destination_stride, 195
ags_linear_interpolate_util_set_factor, 197
ags_linear_interpolate_util_set_format, 196
ags_linear_interpolate_util_set_samplerate, 197
ags_linear_interpolate_util_set_source, 193
ags_linear_interpolate_util_set_source_stride, 193
AgsLinearInterpolateUtil, 201
AGS_LINK_CHANNEL, 2338
AGS_LINK_CHANNEL_CLASS, 2338
AGS_LINK_CHANNEL_GET_CLASS, 2338
ags_link_channel_get_type, 2338
ags_link_channel_new, 2337
AgsLinkChannel, 2339
AgsLinkChannel:channel, 2339
AgsLinkChannel:error, 2339
AgsLinkChannel:link, 2339
AgsLinkChannelClass, 2339
AGS_LV2_ATOM_EVENT, 100
AGS_LV2_ATOM_SEQUENCE, 99
AGS_LV2_CACHE_TURTLE, 94
ags_lv2_cache_turtle_alloc, 94
ags_lv2_cache_turtle_find, 95
ags_lv2_cache_turtle_free, 94
AGS_LV2_CONVERSION, 65
AGS_LV2_CONVERSION_CLASS, 66
AGS_LV2_CONVERSION_DEFAULT_LOWER, 66
AGS_LV2_CONVERSION_DEFAULT_STEP_COUNT, 66
AGS_LV2_CONVERSION_DEFAULT_UPPER, 66
ags_lv2_conversion_flags_get_type, 66
AGS_LV2_CONVERSION_GET_CLASS, 66

ags_lv2_conversion_get_type, 66
ags_lv2_conversion_new, 65
ags_lv2_conversion_set_flags, 64
ags_lv2_conversion_test_flags, 64
ags_lv2_conversion_unset_flags, 65
AGS_LV2_EVENT, 99
AGS_LV2_EVENT_BUFFER, 99
AGS_LV2_EVENT_DATA, 99
AGS_LV2_EVENT_MANAGER, 69
AGS_LV2_EVENT_MANAGER_CLASS, 69
AGS_LV2_EVENT_MANAGER_GET_CLASS, 69
ags_lv2_event_manager_get_instance, 69
ags_lv2_event_manager_get_type, 70
ags_lv2_event_manager_lv2_event_ref, 68
ags_lv2_event_manager_lv2_event_unref, 68
ags_lv2_event_manager_new, 69
AGS_LV2_LOG_MANAGER, 73
AGS_LV2_LOG_MANAGER_CLASS, 73
AGS_LV2_LOG_MANAGER_GET_CLASS, 73
ags_lv2_log_manager_get_instance, 72
ags_lv2_log_manager_get_type, 73
ags_lv2_log_manager_new, 72
ags_lv2_log_manager_printf, 71
ags_lv2_log_manager_vprintf, 71
AGS_LV2_MANAGER, 80
AGS_LV2_MANAGER_CLASS, 80
AGS_LV2_MANAGER_DEFAULT_LOCALE, 80
ags_lv2_manager_find_lv2_plugin, 76
ags_lv2_manager_find_lv2_plugin_with_fallback, 77
AGS_LV2_MANAGER_GET_CLASS, 80
ags_lv2_manager_get_default_path, 75
ags_lv2_manager_get_filenames, 76
ags_lv2_manager_get_instance, 79
ags_lv2_manager_get_lv2_plugin, 79
AGS_LV2_MANAGER_GET_OBJ_MUTEX, 75
ags_lv2_manager_get_type, 80
ags_lv2_manager_global_get_parse_names, 75
ags_lv2_manager_global_get_preserve_turtle, 75
ags_lv2_manager_load_blacklist, 77
ags_lv2_manager_load_default_directory, 78
ags_lv2_manager_load_file, 77
ags_lv2_manager_load_preset, 78
ags_lv2_manager_new, 79
ags_lv2_manager_quick_scan_default_directory, 78
ags_lv2_manager_set_default_path, 75
AGS_LV2_OPTION_MANAGER, 86
AGS_LV2_OPTION_MANAGER_CLASS, 87
AGS_LV2_OPTION_MANAGER_GET_CLASS, 87
ags_lv2_option_manager_get_instance, 86
AGS_LV2_OPTION_MANAGER_GET_OBJ_MUTEX, 82
ags_lv2_option_manager_get_option, 84
ags_lv2_option_manager_get_type, 87
ags_lv2_option_manager_lv2_options_get, 85
ags_lv2_option_manager_lv2_options_set, 85
ags_lv2_option_manager_new, 86
ags_lv2_option_manager_ressource_insert, 83
ags_lv2_option_manager_ressource_lookup, 84
ags_lv2_option_manager_ressource_lookup_extended, 84
ags_lv2_option_manager_ressource_remove, 83
ags_lv2_option_manager_set_option, 85
AGS_LV2_OPTION_RESSOURCE, 82
ags_lv2_option_ressource_alloc, 82
AGS_LV2_OPTIONS_OPTION, 82
AGS_LV2_PLUGIN, 107
ags_lv2_plugin_alloc_atom_sequence, 103
ags_lv2_plugin_atom_sequence_append_midi, 104
ags_lv2_plugin_atom_sequence_free, 103
ags_lv2_plugin_atom_sequence_remove_midi, 104
ags_lv2_plugin_change_program, 106
AGS_LV2_PLUGIN_CLASS, 107
ags_lv2_plugin_clear_atom_sequence, 105
ags_lv2_plugin_clear_event_buffer, 103
AGS_LV2_PLUGIN_DESCRIPTOR, 99
ags_lv2_plugin_event_buffer_alloc, 101
ags_lv2_plugin_event_buffer_append_midi, 102
ags_lv2_plugin_event_buffer_concat, 102
ags_lv2_plugin_event_buffer_free, 101
ags_lv2_plugin_event_buffer_realloc_data, 101
ags_lv2_plugin_event_buffer_remove_midi, 102
ags_lv2_plugin_find_pname, 105
ags_lv2_plugin_find_uri, 105
ags_lv2_plugin_flags_get_type, 107
AGS_LV2_PLUGIN_GET_CLASS, 107
ags_lv2_plugin_get_type, 107
ags_lv2_plugin_new, 106
ags_lv2_plugin_set_flags, 100
ags_lv2_plugin_test_flags, 100
ags_lv2_plugin_unset_flags, 100
AGS_LV2_PORT_PRESET, 115
ags_lv2_port_preset_alloc, 115
ags_lv2_port_preset_free, 115
AGS_LV2_PRESET, 117
AGS_LV2_PRESET_CLASS, 117
ags_lv2_preset_find_preset_label, 116
ags_lv2_preset_find_preset_uri, 116
AGS_LV2_PRESET_GET_CLASS, 117
AGS_LV2_PRESET_GET_OBJ_MUTEX, 115
ags_lv2_preset_get_type, 117
AGS_LV2_PRESET_MANAGER, 113
AGS_LV2_PRESET_MANAGER_CLASS, 113
AGS_LV2_PRESET_MANAGER_GET_CLASS, 113
ags_lv2_preset_manager_get_instance, 112
AGS_LV2_PRESET_MANAGER_GET_OBJ_MUTEX, 112
ags_lv2_preset_manager_get_type, 113
ags_lv2_preset_manager_new, 112
ags_lv2_preset_new, 117
ags_lv2_preset_parse_turtle, 115
AGS_LV2_TURTLE_PARSER, 91
AGS_LV2_TURTLE_PARSER_CLASS, 91
AGS_LV2_TURTLE_PARSER_GET_CLASS, 91
AGS_LV2_TURTLE_PARSER_GET_OBJ_MUTEX, 90
ags_lv2_turtle_parser_get_type, 91
ags_lv2_turtle_parser_new, 90
ags_lv2_turtle_parser_parse, 90

ags_lv2_turtle_parser_parse_names, 90
AGS_LV2_TURTLE_SCANNER, 96
AGS_LV2_TURTLE_SCANNER_CLASS, 96
AGS_LV2_TURTLE_SCANNER_GET_CLASS, 96
AGS_LV2_TURTLE_SCANNER_GET_OBJ_MUTEX, 94
ags_lv2_turtle_scanner_get_type, 96
ags_lv2_turtle_scanner_new, 95
ags_lv2_turtle_scanner_quick_scan, 95
ags_lv2_turtle_scanner_quick_scan_see_also, 95
ags_lv2_turtle_scanner_status_flags_get_type, 96
AGS_LV2_URI_MAP_MANAGER, 123
AGS_LV2_URI_MAP_MANAGER_CLASS, 123
AGS_LV2_URI_MAP_MANAGER_GET_CLASS, 123
ags_lv2_uri_map_manager_get_instance, 122
AGS_LV2_URI_MAP_MANAGER_GET_OBJ_MUTEX, 121
ags_lv2_uri_map_manager_get_type, 123
ags_lv2_uri_map_manager_insert, 121
ags_lv2_uri_map_manager_load_default, 122
ags_lv2_uri_map_manager_lookup, 122
ags_lv2_uri_map_manager_new, 122
ags_lv2_uri_map_manager_remove, 121
ags_lv2_uri_map_manager_uri_to_id, 122
AGS_LV2_URID_MANAGER, 127
AGS_LV2_URID_MANAGER_CLASS, 127
AGS_LV2_URID_MANAGER_GET_CLASS, 127
ags_lv2_urid_manager_get_instance, 126
AGS_LV2_URID_MANAGER_GET_OBJ_MUTEX, 125
ags_lv2_urid_manager_get_type, 127
ags_lv2_urid_manager_insert, 125
ags_lv2_urid_manager_load_default, 126
ags_lv2_urid_manager_lookup, 126
ags_lv2_urid_manager_map, 126
ags_lv2_urid_manager_new, 127
ags_lv2_urid_manager_remove, 125
ags_lv2_urid_manager_unmap, 126
AGS_LV2_WORKER, 132
ags_lv2_worker_alloc_response_data, 130
AGS_LV2_WORKER_CLASS, 133
ags_lv2_worker_do_poll, 132
ags_lv2_worker_flags_get_type, 133
ags_lv2_worker_free_response_data, 131
AGS_LV2_WORKER_GET_CLASS, 133
AGS_LV2_WORKER_GET_OBJ_MUTEX, 129
ags_lv2_worker_get_type, 133
AGS_LV2_WORKER_MANAGER, 137
AGS_LV2_WORKER_MANAGER_CLASS, 137
AGS_LV2_WORKER_MANAGER_GET_CLASS, 137
ags_lv2_worker_manager_get_instance, 136
AGS_LV2_WORKER_MANAGER_GET_OBJ_MUTEX, 136
ags_lv2_worker_manager_get_type, 137
ags_lv2_worker_manager_new, 136
ags_lv2_worker_manager_pull_worker, 136
ags_lv2_worker_new, 132
ags_lv2_worker_respond, 131
AGS_LV2_WORKER_RESPONSE_DATA, 129
ags_lv2_worker_schedule_work, 131
ags_lv2_worker_set_flags, 130
ags_lv2_worker_test_flags, 129
ags_lv2_worker_unset_flags, 130
AgsLv2CacheTurtle, 97
AgsLv2Conversion, 67
AgsLv2Conversion:lower, 67
AgsLv2Conversion:step-count, 67
AgsLv2Conversion:upper, 67
AgsLv2ConversionClass, 67
AgsLv2ConversionFlags, 66
AgsLv2EventManager, 70
AgsLv2EventManagerClass, 70
AgsLv2LogManager, 73
AgsLv2LogManagerClass, 73
AgsLv2Manager, 80
AgsLv2Manager:locale, 81
AgsLv2ManagerClass, 81
AgsLv2OptionManager, 87
AgsLv2OptionManager::get-option, 88
AgsLv2OptionManager::set-option, 88
AgsLv2OptionManagerClass, 87
AgsLv2OptionRessource, 87
AgsLv2Plugin, 108
AgsLv2Plugin::change-program, 110
AgsLv2Plugin:doap-name, 108
AgsLv2Plugin:foaf-homepage, 109
AgsLv2Plugin:foaf-mbox, 109
AgsLv2Plugin:foaf-name, 109
AgsLv2Plugin:manifest, 109
AgsLv2Plugin:pname, 109
AgsLv2Plugin:preset, 110
AgsLv2Plugin:turtle, 110
AgsLv2Plugin:ui-uri, 110
AgsLv2Plugin:uri, 110
AgsLv2PluginClass, 108
AgsLv2PluginFlags, 107
AgsLv2PortPreset, 118
AgsLv2Preset, 118
AgsLv2Preset:applies-to, 118
AgsLv2Preset:bank, 118
AgsLv2Preset:lv2-plugin, 119
AgsLv2Preset:manifest, 119
AgsLv2Preset:port-preset, 119
AgsLv2Preset:preset-label, 119
AgsLv2Preset:turtle, 119
AgsLv2Preset:uri, 120
AgsLv2PresetClass, 118
AgsLv2PresetManager, 113
AgsLv2PresetManagerClass, 113
AgsLv2TurtleParser, 92
AgsLv2TurtleParser:plugin, 92
AgsLv2TurtleParser:preset, 92
AgsLv2TurtleParser:turtle, 92
AgsLv2TurtleParser:ui-plugin, 93
AgsLv2TurtleParserClass, 92
AgsLv2TurtleScanner, 97
AgsLv2TurtleScannerClass, 97
AgsLv2TurtleScannerStatusFlags, 96

AGS_LV2UI_MANAGER, 142
AGS_LV2UI_MANAGER_CLASS, 142
ags_lv2ui_manager_find_lv2ui_plugin, 140
ags_lv2ui_manager_find_lv2ui_plugin_with_index, 140
AGS_LV2UI_MANAGER_GET_CLASS, 142
ags_lv2ui_manager_get_default_path, 139
ags_lv2ui_manager_get_filenames, 139
ags_lv2ui_manager_get_instance, 141
AGS_LV2UI_MANAGER_GET_OBJ_MUTEX, 139
ags_lv2ui_manager_get_type, 142
ags_lv2ui_manager_load_blacklist, 140
ags_lv2ui_manager_load_default_directory, 141
ags_lv2ui_manager_load_file, 141
ags_lv2ui_manager_new, 142
ags_lv2ui_manager_set_default_path, 139
AGS_LV2UI_PLUGIN, 146
AGS_LV2UI_PLUGIN_CLASS, 146
AGS_LV2UI_PLUGIN_DESCRIPTOR, 144
ags_lv2ui_plugin_find_gui_uri, 145
ags_lv2ui_plugin_flags_get_type, 147
AGS_LV2UI_PLUGIN_GET_CLASS, 146
ags_lv2ui_plugin_get_type, 147
ags_lv2ui_plugin_new, 146
ags_lv2ui_plugin_set_flags, 145
ags_lv2ui_plugin_test_flags, 144
ags_lv2ui_plugin_unset_flags, 145
AgsLv2uiManager, 143
AgsLv2uiManagerClass, 143
AgsLv2uiPlugin, 147
AgsLv2uiPlugin:gui-turtle, 148
AgsLv2uiPlugin:gui-uri, 148
AgsLv2uiPlugin:lv2-plugin, 148
AgsLv2uiPlugin:manifest, 148
AgsLv2uiPluginClass, 147
AgsLv2uiPluginFlags, 147
AgsLv2UridManager, 128
AgsLv2UridManagerClass, 128
AgsLv2UriMapManager, 123
AgsLv2UriMapManagerClass, 124
AgsLv2Worker, 133
AgsLv2Worker:handle, 134
AgsLv2Worker:response-data, 134
AgsLv2Worker:work-data, 134
AgsLv2Worker:work-size, 134
AgsLv2Worker:worker-thread, 135
AgsLv2WorkerClass, 134
AgsLv2WorkerFlags, 133
AgsLv2WorkerManager, 137
AgsLv2WorkerManagerClass, 137
AgsLv2WorkerResponseData, 133

M

ags_mid_file_read_midi, 1563
AGS_MIDI, 1281
AgsMidi, 1283
AgsMidi:audio, 1283
AgsMidi:audio-channel, 1284
AgsMidi:timestamp, 1284
AgsMidi:track, 1284
ags_midi_add, 1280
ags_midi_add_track, 1280
ags_midi_buffer_util_decode, 1551
ags_midi_buffer_util_get_change_parameter, 1536
ags_midi_buffer_util_get_change_pressure, 1539
ags_midi_buffer_util_get_change_program, 1538
ags_midi_buffer_util_get_end_of_track, 1550
ags_midi_buffer_util_get_header, 1532
ags_midi_buffer_util_get_int16, 1530
ags_midi_buffer_util_get_int24, 1530
ags_midi_buffer_util_get_int32, 1531
ags_midi_buffer_util_get_key_off, 1534
ags_midi_buffer_util_get_key_on, 1533
ags_midi_buffer_util_get_key_pressure, 1535
ags_midi_buffer_util_get_key_signature, 1548
ags_midi_buffer_util_get_pitch_bend, 1537
ags_midi_buffer_util_get_quarter_frame, 1541
ags_midi_buffer_util_get_sequence_number, 1544
ags_midi_buffer_util_get_sequencer_meta_event, 1549
ags_midi_buffer_util_get_smtpe, 1545
ags_midi_buffer_util_get_song_position, 1542
ags_midi_buffer_util_get_song_select, 1543
ags_midi_buffer_util_get_sysex, 1540
ags_midi_buffer_util_get_tempo, 1546
ags_midi_buffer_util_get_text_event, 1550
ags_midi_buffer_util_get_time_signature, 1547
ags_midi_buffer_util_get_track, 1533
ags_midi_buffer_util_get_tune_request, 1543
ags_midi_buffer_util_get_type, 1552
ags_midi_buffer_util_get_varlength, 1529
ags_midi_buffer_util_get_varlength_size, 1529
ags_midi_buffer_util_put_change_parameter, 1536
ags_midi_buffer_util_put_change_pressure, 1539
ags_midi_buffer_util_put_change_program, 1538
ags_midi_buffer_util_put_end_of_track, 1550
ags_midi_buffer_util_put_header, 1531
ags_midi_buffer_util_put_int16, 1530
ags_midi_buffer_util_put_int24, 1530
ags_midi_buffer_util_put_int32, 1531
ags_midi_buffer_util_put_key_off, 1534
ags_midi_buffer_util_put_key_on, 1533
ags_midi_buffer_util_put_key_pressure, 1535
ags_midi_buffer_util_put_key_signature, 1547
ags_midi_buffer_util_put_pitch_bend, 1537
ags_midi_buffer_util_put_quarter_frame, 1541
ags_midi_buffer_util_put_sequence_number, 1544
ags_midi_buffer_util_put_sequencer_meta_event, 1548
ags_midi_buffer_util_put_smtpe, 1545
ags_midi_buffer_util_put_song_position, 1541
ags_midi_buffer_util_put_song_select, 1542
ags_midi_buffer_util_put_sysex, 1540
ags_midi_buffer_util_put_tempo, 1546
ags_midi_buffer_util_put_text_event, 1549
ags_midi_buffer_util_put_time_signature, 1546
ags_midi_buffer_util_put_track, 1532

ags_midi_buffer_util_put_tune_request, 1543
ags_midi_buffer_util_put_varlength, 1529
ags_midi_buffer_util_seek_message, 1551
AGS_MIDI_BUILDER, 1594
ags_midi_builder_append_change_parameter, 1585
ags_midi_builder_append_change_pitch_bend, 1585
ags_midi_builder_append_change_pressure, 1586
ags_midi_builder_append_change_program, 1586
ags_midi_builder_append_header, 1583
ags_midi_builder_append_key_off, 1584
ags_midi_builder_append_key_on, 1584
ags_midi_builder_append_key_pressure, 1585
ags_midi_builder_append_key_signature, 1590
ags_midi_builder_append_quarter_frame, 1587
ags_midi_builder_append_sequence_number, 1588
ags_midi_builder_append_sequencer_meta_event, 1590
ags_midi_builder_append_smtpe, 1589
ags_midi_builder_append_song_position, 1587
ags_midi_builder_append_song_select, 1588
ags_midi_builder_append_sysex, 1587
ags_midi_builder_append_tempo, 1589
ags_midi_builder_append_text_event, 1591
ags_midi_builder_append_time_signature, 1590
ags_midi_builder_append_track, 1583
ags_midi_builder_append_tune_request, 1588
ags_midi_builder_append_xml_node, 1591
ags_midi_builder_build, 1592
AGS_MIDI_BUILDER_CLASS, 1594
ags_midi_builder_from_xml_doc, 1591
AGS_MIDI_BUILDER_GET_CLASS, 1594
ags_midi_builder_get_data, 1592
ags_midi_builder_get_data_with_length, 1593
AGS_MIDI_BUILDER_GET_OBJ_MUTEX, 1580
ags_midi_builder_get_type, 1594
AGS_MIDI_BUILDER_HEADER, 1580
ags_midi_builder_header_alloc, 1580
ags_midi_builder_header_free, 1580
ags_midi_builder_midi_putc, 1582
ags_midi_builder_new, 1593
ags_midi_builder_new_from_filename, 1593
ags_midi_builder_on_error, 1583
ags_midi_builder_open_filename, 1592
AGS_MIDI_BUILDER_TRACK, 1580
ags_midi_builder_track_alloc, 1581
ags_midi_builder_track_find_delta_time_with_track_name, 1581
ags_midi_builder_track_free, 1581
ags_midi_builder_track_get_delta_time_offset, 1582
ags_midi_builder_track_insert_midi_message, 1582
ags_midi_builder_write, 1593
AGS_MIDI_CLASS, 1281
AGS_MIDI_CLIPBOARD_FORMAT, 1283
AGS_MIDI_CLIPBOARD_TYPE, 1283
AGS_MIDI_CLIPBOARD_VERSION, 1282
AGS_MIDI_DEFAULT_BPM, 1282
AGS_MIDI_DEFAULT_DURATION, 1282
AGS_MIDI_DEFAULT_JIFFIE, 1282
AGS_MIDI_DEFAULT_LENGTH, 1282
AGS_MIDI_DEFAULT_OFFSET, 1282
AGS_MIDI_EVENT, 1619
AGS_MIDI_FILE, 1563
AGS_MIDI_FILE_CLASS, 1563
ags_midi_file_close, 1556
AGS_MIDI_FILE_DEFAULT_BEATS, 1564
AGS_MIDI_FILE_DEFAULT_FORMAT, 1564
AGS_MIDI_FILE_DEFAULT_FPS, 1564
AGS_MIDI_FILE_DEFAULT_OFFSET, 1564
AGS_MIDI_FILE_DEFAULT_TICKS, 1564
AGS_MIDI_FILE_ERROR, 1565
ags_midi_file_error_quark, 1554
ags_midi_file_flush, 1557
AGS_MIDI_FILE_GET_CLASS, 1563
AGS_MIDI_FILE_GET_OBJ_MUTEX, 1554
ags_midi_file_get_type, 1563
AGS_MIDI_FILE_MAX_TEXT_LENGTH, 1564
AGS_MIDI_FILE_MTHD, 1564
AGS_MIDI_FILE_MTRK, 1564
ags_midi_file_new, 1563
ags_midi_file_open, 1554
ags_midi_file_open_from_data, 1555
ags_midi_file_read, 1556
ags_midi_file_read_byte, 1557
ags_midi_file_read_gint16, 1558
ags_midi_file_read_gint24, 1558
ags_midi_file_read_gint32, 1558
ags_midi_file_read_header, 1561
ags_midi_file_read_notation, 1562
ags_midi_file_read_text, 1559
ags_midi_file_read_track_data, 1562
ags_midi_file_read_varlength, 1559
ags_midi_file_rw_open, 1555
ags_midi_file_seek, 1557
AGS_MIDI_FILE_TRACK, 1554
ags_midi_file_write, 1556
ags_midi_file_write_byte, 1560
ags_midi_file_write_gint16, 1560
ags_midi_file_write_gint24, 1560
ags_midi_file_write_gint32, 1560
ags_midi_file_write_header, 1562
ags_midi_file_write_text, 1561
ags_midi_file_write_track_data, 1562
ags_midi_file_write_varlength, 1561
ags_midi_find_near_timestamp, 1276
ags_midi_get_audio, 1277
ags_midi_get_audio_channel, 1278
AGS_MIDI_GET_CLASS, 1282
AGS_MIDI_GET_OBJ_MUTEX, 1275
ags_midi_get_obj_mutex, 1275
ags_midi_get_timestamp, 1278
ags_midi_get_track, 1279
ags_midi_get_type, 1282
AGS_MIDI_MINIMUM_TRACK_LENGTH, 1282
ags_midi_new, 1281
AGS_MIDI_PARSER, 1619

ags_midi_parser_change_channel_pressure, 1612
ags_midi_parser_change_parameter, 1611
ags_midi_parser_change_pitch_bend, 1611
ags_midi_parser_change_program, 1612
ags_midi_parser_channel_message, 1609
AGS_MIDI_PARSER_CLASS, 1619
ags_midi_parser_end_of_track, 1614
AGS_MIDI_PARSER_GET_CLASS, 1619
AGS_MIDI_PARSER_GET_OBJ_MUTEX, 1605
ags_midi_parser_get_type, 1619
ags_midi_parser_key_off, 1610
ags_midi_parser_key_on, 1610
ags_midi_parser_key_pressure, 1610
ags_midi_parser_key_signature, 1616
AGS_MIDI_PARSER_MAX_TEXT_LENGTH, 1619
ags_midi_parser_meta_event, 1614
ags_midi_parser_meta_misc, 1616
ags_midi_parser_midi_getc, 1607
AGS_MIDI_PARSER_MTCK, 1619
AGS_MIDI_PARSER_MTHD, 1619
ags_midi_parser_new, 1618
ags_midi_parser_new_from_filename, 1618
ags_midi_parser_not_defined, 1612
ags_midi_parser_on_error, 1608
ags_midi_parser_open_filename, 1617
ags_midi_parser_parse_bytes, 1608
ags_midi_parser_parse_full, 1608
ags_midi_parser_parse_header, 1608
ags_midi_parser_parse_track, 1609
ags_midi_parser_read_gint16, 1605
ags_midi_parser_read_gint24, 1605
ags_midi_parser_read_gint32, 1606
ags_midi_parser_read_text, 1606
ags_midi_parser_read_varlength, 1606
ags_midi_parser_sequence_number, 1614
ags_midi_parser_sequencer_meta_event, 1616
ags_midi_parser_set_buffer, 1617
ags_midi_parser_set_file_length, 1618
ags_midi_parser_smtpe, 1615
ags_midi_parser_sysex, 1613
ags_midi_parser_system_common, 1613
ags_midi_parser_tempo, 1615
ags_midi_parser_text_event, 1617
ags_midi_parser_ticks_to_sec, 1607
ags_midi_parser_time_signature, 1616
ags_midi_remove_track, 1280
ags_midi_set_audio, 1278
ags_midi_set_audio_channel, 1278
ags_midi_set_flags, 1276
ags_midi_set_timestamp, 1279
ags_midi_set_track, 1279
ags_midi_sort_func, 1277
ags_midi_test_flags, 1275
AGS_MIDI_TICS_PER_BEAT, 1282
ags_midi_unset_flags, 1276
ags_midi_util_delta_time_to_offset, 1576
ags_midi_util_get_change_parameter, 1572
ags_midi_util_get_change_pressure, 1574
ags_midi_util_get_change_program, 1573
ags_midi_util_get_key_off, 1571
ags_midi_util_get_key_on, 1571
ags_midi_util_get_key_pressure, 1572
ags_midi_util_get_pitch_bend, 1573
ags_midi_util_get_quarter_frame, 1575
ags_midi_util_get_song_position, 1575
ags_midi_util_get_song_select, 1575
ags_midi_util_get_sysex, 1574
ags_midi_util_get_type, 1577
ags_midi_util_is_change_parameter, 1568
ags_midi_util_is_change_pressure, 1569
ags_midi_util_is_change_program, 1568
ags_midi_util_is_key_off, 1567
ags_midi_util_is_key_on, 1567
ags_midi_util_is_key_pressure, 1568
ags_midi_util_is_meta_event, 1571
ags_midi_util_is_pitch_bend, 1568
ags_midi_util_is_quarter_frame, 1569
ags_midi_util_is_song_position, 1570
ags_midi_util_is_song_select, 1570
ags_midi_util_is_sysex, 1569
ags_midi_util_is_tune_request, 1570
ags_midi_util_offset_to_delta_time, 1577
ags_midi_util_to_smf, 1576
AgsMidiBufferUtil, 1552
AgsMidiBufferUtilMtcQuarterFrameMessageType, 1552
AgsMidiBufferUtilSmtpeFrameRate, 1552
AgsMidiBuilder, 1595
AgsMidiBuilder::append-header, 1597
AgsMidiBuilder::append-track, 1598
AgsMidiBuilder::change-channel-pressure, 1598
AgsMidiBuilder::change-parameter, 1598
AgsMidiBuilder::change-pitch-bend, 1599
AgsMidiBuilder::change-program, 1599
AgsMidiBuilder::key-off, 1599
AgsMidiBuilder::key-on, 1599
AgsMidiBuilder::key-pressure, 1599
AgsMidiBuilder::key-signature, 1600
AgsMidiBuilder::midi-putc, 1600
AgsMidiBuilder::on-error, 1600
AgsMidiBuilder::quarter-frame, 1601
AgsMidiBuilder::sequence-number, 1601
AgsMidiBuilder::sequencer-meta-event, 1601
AgsMidiBuilder::smtpe, 1601
AgsMidiBuilder::song-position, 1601
AgsMidiBuilder::song-select, 1602
AgsMidiBuilder::sysex, 1602
AgsMidiBuilder::tempo, 1602
AgsMidiBuilder::text-event, 1602
AgsMidiBuilder::time-signature, 1602
AgsMidiBuilder::tune-request, 1603
AgsMidiBuilder:file, 1597
AgsMidiBuilderClass, 1595
AgsMidiBuilderFlags, 1594
AgsMidiBuilderHeader, 1595

AgsMidiBuilderTrack, 1595
AgsMidiChunkFlags, 1620
AgsMidiClass, 1283
AgsMidiFile, 1565
AgsMidiFile:filename, 1566
AgsMidiFileClass, 1565
AgsMidiFileError, 1565
AgsMidiFileFlags, 1565
AgsMidiFileStatus, 1564
AgsMidiFileTrack, 1565
AgsMidiFlags, 1283
AgsMidiParser, 1620
AgsMidiParser::change-channel-pressure, 1621
AgsMidiParser::change-parameter, 1621
AgsMidiParser::change-pitch-bend, 1622
AgsMidiParser::change-program, 1622
AgsMidiParser::end-of-track, 1623
AgsMidiParser::key-off, 1623
AgsMidiParser::key-on, 1624
AgsMidiParser::key-pressure, 1624
AgsMidiParser::key-signature, 1625
AgsMidiParser::meta-event, 1625
AgsMidiParser::midi-getc, 1626
AgsMidiParser::not-defined, 1626
AgsMidiParser::on-error, 1627
AgsMidiParser::parse-bytes, 1627
AgsMidiParser::parse-full, 1628
AgsMidiParser::parse-header, 1628
AgsMidiParser::parse-track, 1629
AgsMidiParser::sequence-number, 1629
AgsMidiParser::sequencer-meta-event, 1629
AgsMidiParser::smtpe, 1630
AgsMidiParser::sysex, 1630
AgsMidiParser::system-common, 1631
AgsMidiParser::tempo, 1631
AgsMidiParser::text-event, 1632
AgsMidiParser::time-signature, 1632
AgsMidiParser:file, 1621
AgsMidiParserClass, 1620
AgsMidiParserFlags, 1620
AgsMidiUtil, 1577
AGS_MOVE_NOTE, 2342
AGS_MOVE_NOTE_CLASS, 2342
AGS_MOVE_NOTE_DEFAULT_X_LENGTH, 2342
AGS_MOVE_NOTE_DEFAULT_Y_LENGTH, 2342
AGS_MOVE_NOTE_GET_CLASS, 2342
ags_move_note_get_type, 2342
ags_move_note_new, 2341
ags_move_note_set_selection, 2340
AgsMoveNote, 2342
AgsMoveNote:absolute, 2343
AgsMoveNote:audio, 2343
AgsMoveNote:first-x, 2343
AgsMoveNote:first-y, 2343
AgsMoveNote:move-x, 2344
AgsMoveNote:move-y, 2344
AgsMoveNote:notation, 2344
AgsMoveNote:relative, 2344
AgsMoveNote:selection, 2345
AgsMoveNoteClass, 2343

N

ags_noise_util_alloc, 354
ags_noise_util_compute, 363
ags_noise_util_compute_complex, 363
ags_noise_util_compute_double, 363
ags_noise_util_compute_float, 363
ags_noise_util_compute_s16, 362
ags_noise_util_compute_s24, 362
ags_noise_util_compute_s32, 362
ags_noise_util_compute_s64, 362
ags_noise_util_compute_s8, 361
ags_noise_util_copy, 354
AGS_NOISE_UTIL_DEFAULT_FREQUENCY, 364
ags_noise_util_free, 354
ags_noise_util_get_buffer_length, 357
ags_noise_util_get_destination, 356
ags_noise_util_get_destination_stride, 356
ags_noise_util_get_format, 358
ags_noise_util_get_frame_count, 360
ags_noise_util_get_frequency, 360
ags_noise_util_get_offset, 361
ags_noise_util_get_samplerate, 358
ags_noise_util_get_source, 355
ags_noise_util_get_source_stride, 355
ags_noise_util_get_type, 364
AGS_NOISE_UTIL_GET_UNSCALED_VALUE, 354
AGS_NOISE_UTIL_GET_VALUE, 354
ags_noise_util_get_volume, 359
AGS_NOISE_UTIL_RESET, 353
ags_noise_util_set_buffer_length, 357
ags_noise_util_set_destination, 356
ags_noise_util_set_destination_stride, 357
ags_noise_util_set_format, 358
ags_noise_util_set_frame_count, 361
ags_noise_util_set_frequency, 360
ags_noise_util_set_offset, 361
ags_noise_util_set_samplerate, 359
ags_noise_util_set_source, 355
ags_noise_util_set_source_stride, 356
ags_noise_util_set_volume, 359
AgsNoiseUtil, 364
AgsNoiseuUtilMode, 364
AGS_NOTATION, 1181
AgsNotation, 1183
AgsNotation:audio, 1183
AgsNotation:audio-channel, 1184
AgsNotation:note, 1184
AgsNotation:timestamp, 1184
ags_notation_add, 1167
ags_notation_add_all_to_selection, 1177
ags_notation_add_note, 1172
ags_notation_add_point_to_selection, 1175
ags_notation_add_region_to_selection, 1176

AGS_NOTATION_CLASS, 1181
AGS_NOTATION_CLIPBOARD_FORMAT, 1183
AGS_NOTATION_CLIPBOARD_TYPE, 1182
AGS_NOTATION_CLIPBOARD_VERSION, 1182
ags_notation_copy_selection, 1177
ags_notation_cut_selection, 1178
AGS_NOTATION_DEFAULT_BPM, 1181
AGS_NOTATION_DEFAULT_DIVISION, 1182
AGS_NOTATION_DEFAULT_DURATION, 1182
AGS_NOTATION_DEFAULT_END, 1182
AGS_NOTATION_DEFAULT_JIFFIE, 1182
AGS_NOTATION_DEFAULT_LENGTH, 1182
AGS_NOTATION_DEFAULT_OFFSET, 1182
ags_notation_find_near_timestamp, 1166
ags_notation_find_offset, 1174
ags_notation_find_point, 1173
ags_notation_find_region, 1174
ags_notation_flags_get_type, 1181
ags_notation_free_all_selection, 1175
ags_notation_free_selection, 1175
ags_notation_from_raw_midi, 1180
ags_notation_get_audio, 1168
ags_notation_get_audio_channel, 1168
AGS_NOTATION_GET_CLASS, 1181
ags_notation_get_is_minor, 1169
ags_notation_get_note, 1171
AGS_NOTATION_GET_OBJ_MUTEX, 1165
ags_notation_get_obj_mutex, 1165
ags_notation_get_selection, 1173
ags_notation_get_sharp_flats, 1170
ags_notation_get_timestamp, 1170
ags_notation_get_type, 1181
ags_notation_insert_from_clipboard, 1178
ags_notation_insert_from_clipboard_extended, 1178
ags_notation_is_note_selected, 1173
AGS_NOTATION_MAXIMUM_NOTE_LENGTH, 1182
AGS_NOTATION_MINIMUM_NOTE_LENGTH, 1182
ags_notation_new, 1180
ags_notation_remove_note, 1172
ags_notation_remove_note_at_position, 1172
ags_notation_remove_point_from_selection, 1176
ags_notation_remove_region_from_selection, 1177
ags_notation_set_audio, 1168
ags_notation_set_audio_channel, 1169
ags_notation_set_flags, 1166
ags_notation_set_is_minor, 1169
ags_notation_set_note, 1171
ags_notation_set_sharp_flats, 1170
ags_notation_set_timestamp, 1171
ags_notation_sort_func, 1167
ags_notation_test_flags, 1165
AGS_NOTATION_TICS_PER_BEAT, 1181
ags_notation_to_raw_midi, 1179
ags_notation_unset_flags, 1166
AgsNotationClass, 1183
AgsNotationFlags, 1183
AGS_NOTE, 1203
AgsNote, 1204
AgsNote:attack, 1205
AgsNote:decay, 1205
AgsNote:frequency, 1205
AgsNote:note-name, 1205
AgsNote:ratio, 1206
AgsNote:release, 1206
AgsNote:rt-attack, 1206
AgsNote:rt-offset, 1206
AgsNote:stream-attack, 1206
AgsNote:stream-delay, 1207
AgsNote:stream-frame-count, 1207
AgsNote:sustain, 1207
AgsNote:x0, 1207
AgsNote:x1, 1207
AgsNote:y, 1208
AGS_NOTE_CLASS, 1203
ags_note_duplicate, 1202
ags_note_find_next, 1196
ags_note_find_prev, 1196
ags_note_from_raw_midi, 1200
ags_note_from_raw_midi_extended, 1200
ags_note_from_seq_event, 1201
ags_note_from_seq_event_extended, 1201
ags_note_get_attack, 1192
AGS_NOTE_GET_CLASS, 1204
ags_note_get_decay, 1194
ags_note_get_is_minor, 1188
AGS_NOTE_GET_OBJ_MUTEX, 1186
ags_note_get_obj_mutex, 1186
ags_note_get_ratio, 1195
ags_note_get_release, 1194
ags_note_get_rt_attack, 1192
ags_note_get_rt_offset, 1191
ags_note_get_sharp_flats, 1189
ags_note_get_sustain, 1193
ags_note_get_type, 1204
ags_note_get_x0, 1189
ags_note_get_x1, 1190
ags_note_get_y, 1191
ags_note_length_to_smf_delta_time, 1196
ags_note_new, 1202
ags_note_new_with_offset, 1203
ags_note_set_attack, 1193
ags_note_set_decay, 1194
ags_note_set_flags, 1187
ags_note_set_is_minor, 1188
ags_note_set_ratio, 1195
ags_note_set_release, 1195
ags_note_set_rt_attack, 1192
ags_note_set_rt_offset, 1192
ags_note_set_sharp_flats, 1189
ags_note_set_sustain, 1193
ags_note_set_x0, 1190
ags_note_set_x1, 1190
ags_note_set_y, 1191
ags_note_smf_delta_time_to_length, 1197

ags_note_sort_func, 1188
ags_note_test_flags, 1187
ags_note_to_raw_midi, 1198
ags_note_to_raw_midi_extended, 1198
ags_note_to_seq_event, 1199
ags_note_to_seq_event_extended, 1199
ags_note_unset_flags, 1187
AgsNoteClass, 1205
AgsNoteFlags, 1204

O

AGS_OPEN_FILE, 2346
AGS_OPEN_FILE_CLASS, 2347
AGS_OPEN_FILE_GET_CLASS, 2347
ags_open_file_get_type, 2347
ags_open_file_new, 2346
AGS_OPEN_SF2_INSTRUMENT, 2353
AGS_OPEN_SF2_INSTRUMENT_CLASS, 2353
AGS_OPEN_SF2_INSTRUMENT_GET_CLASS, 2353
ags_open_sf2_instrument_get_type, 2353
ags_open_sf2_instrument_new, 2352
AGS_OPEN_SF2_SAMPLE, 2357
AGS_OPEN_SF2_SAMPLE_CLASS, 2357
AGS_OPEN_SF2_SAMPLE_GET_CLASS, 2357
ags_open_sf2_sample_get_type, 2357
ags_open_sf2_sample_new, 2356
AGS_OPEN_SFZ_FILE, 2361
AGS_OPEN_SFZ_FILE_CLASS, 2361
AGS_OPEN_SFZ_FILE_GET_CLASS, 2361
ags_open_sfz_file_get_type, 2361
ags_open_sfz_file_new, 2360
AGS_OPEN_SINGLE_FILE, 2350
AGS_OPEN_SINGLE_FILE_CLASS, 2350
AGS_OPEN_SINGLE_FILE_GET_CLASS, 2350
ags_open_single_file_get_type, 2350
ags_open_single_file_new, 2349
AGS_OPEN_WAVE, 2364
AGS_OPEN_WAVE_CLASS, 2364
AGS_OPEN_WAVE_GET_CLASS, 2364
ags_open_wave_get_type, 2365
ags_open_wave_new, 2364
AgsOpenFile, 2347
AgsOpenFile:audio, 2347
AgsOpenFile:create-channels, 2348
AgsOpenFile:filename, 2348
AgsOpenFile:overwrite-channels, 2348
AgsOpenFileClass, 2347
AgsOpenSf2Instrument, 2354
AgsOpenSf2Instrument:audio, 2354
AgsOpenSf2Instrument:filename, 2354
AgsOpenSf2Instrument:instrument, 2354
AgsOpenSf2Instrument:ipatch, 2355
AgsOpenSf2Instrument:preset, 2355
AgsOpenSf2Instrument:start-pad, 2355
AgsOpenSf2InstrumentClass, 2354
AgsOpenSf2Sample, 2358
AgsOpenSf2Sample:audio-channel, 2358
AgsOpenSf2Sample:channel, 2358
AgsOpenSf2Sample:filename, 2358
AgsOpenSf2Sample:instrument, 2359
AgsOpenSf2Sample:ipatch-sample, 2359
AgsOpenSf2Sample:preset, 2359
AgsOpenSf2Sample:sample, 2359
AgsOpenSf2SampleClass, 2358
AgsOpenSFZFile, 2362
AgsOpenSFZFile:audio, 2362
AgsOpenSFZFile:filename, 2362
AgsOpenSFZFile:sfz-file, 2362
AgsOpenSFZFile:start-pad, 2362
AgsOpenSFZFileClass, 2362
AgsOpenSingleFile, 2350
AgsOpenSingleFile:audio-channel, 2351
AgsOpenSingleFile:channel, 2351
AgsOpenSingleFile:filename, 2351
AgsOpenSingleFileClass, 2351
AgsOpenWave, 2365
AgsOpenWave:audio, 2365
AgsOpenWave:audio-file, 2365
AgsOpenWave:filename, 2366
AgsOpenWave:start-pad, 2366
AgsOpenWaveClass, 2365
AGS_OSC_ACTION_CONTROLLER, 1721
AGS_OSC_ACTION_CONTROLLER_CLASS, 1721
AGS_OSC_ACTION_CONTROLLER_GET_CLASS, 1721
ags_osc_action_controller_get_type, 1721
ags_osc_action_controller_new, 1720
ags_osc_action_controller_run_action, 1720
ags_osc_buffer_util_get_blob, 1638
ags_osc_buffer_util_get_bundle, 1645
ags_osc_buffer_util_get_char, 1640
ags_osc_buffer_util_get_double, 1640
ags_osc_buffer_util_get_float, 1637
ags_osc_buffer_util_get_int32, 1635
ags_osc_buffer_util_get_int64, 1639
ags_osc_buffer_util_get_message, 1644
ags_osc_buffer_util_get_midi, 1642
ags_osc_buffer_util_get_packet, 1643
ags_osc_buffer_util_get_rgba, 1641
ags_osc_buffer_util_get_string, 1638
ags_osc_buffer_util_get_timetag, 1636
ags_osc_buffer_util_get_type, 1645
ags_osc_buffer_util_put_blob, 1638
ags_osc_buffer_util_put_bundle, 1644
ags_osc_buffer_util_put_char, 1640
ags_osc_buffer_util_put_double, 1639
ags_osc_buffer_util_put_float, 1637
ags_osc_buffer_util_put_int32, 1635
ags_osc_buffer_util_put_int64, 1639
ags_osc_buffer_util_put_message, 1643
ags_osc_buffer_util_put_midi, 1642
ags_osc_buffer_util_put_packet, 1642
ags_osc_buffer_util_put_packets, 1643
ags_osc_buffer_util_put_rgba, 1641
ags_osc_buffer_util_put_string, 1637

ags_osc_buffer_util_put_timetag, 1636
AGS_OSC_BUILDER, 1659
ags_osc_builder_append_bundle, 1656
ags_osc_builder_append_message, 1657
ags_osc_builder_append_packet, 1656
ags_osc_builder_append_value, 1657
ags_osc_builder_build, 1658
AGS_OSC_BUILDER_BUNDLE, 1653
ags_osc_builder_bundle_alloc, 1654
ags_osc_builder_bundle_free, 1654
AGS_OSC_BUILDER_CLASS, 1659
AGS_OSC_BUILDER_DEFAULT_CHUNK_SIZE, 1660
AGS_OSC_BUILDER_GET_CLASS, 1659
ags_osc_builder_get_data, 1658
ags_osc_builder_get_data_with_length, 1658
AGS_OSC_BUILDER_GET_OBJ_MUTEX, 1653
ags_osc_builder_get_type, 1660
AGS_OSC_BUILDER_MESSAGE, 1653
ags_osc_builder_message_alloc, 1655
AGS_OSC_BUILDER_MESSAGE_DEFAULT_CHUNK_SIZE, 1660
ags_osc_builder_message_free, 1655
ags_osc_builder_new, 1659
ags_osc_builder_on_error, 1656
ags_osc_builder_osc_putc, 1655
AGS_OSC_BUILDER_PACKET, 1653
ags_osc_builder_packet_alloc, 1653
ags_osc_builder_packet_free, 1654
AGS_OSC_CLIENT, 1681
AGS_OSC_CLIENT_CHUNK_SIZE, 1682
AGS_OSC_CLIENT_CLASS, 1681
ags_osc_client_connect, 1680
AGS_OSC_CLIENT_DEFAULT_CACHE_DATA_LENGTH, 1682
AGS_OSC_CLIENT_DEFAULT_DOMAIN, 1682
AGS_OSC_CLIENT_DEFAULT_INET4_ADDRESS, 1682
AGS_OSC_CLIENT_DEFAULT_INET6_ADDRESS, 1682
AGS_OSC_CLIENT_DEFAULT_MAX_ADDRESS_LENGTH, 1682
AGS_OSC_CLIENT_DEFAULT_MAX_RETRY, 1682
AGS_OSC_CLIENT_DEFAULT_SERVER_PORT, 1682
ags_osc_client_flags_get_type, 1682
AGS_OSC_CLIENT_GET_CLASS, 1681
AGS_OSC_CLIENT_GET_OBJ_MUTEX, 1678
ags_osc_client_get_type, 1682
ags_osc_client_new, 1681
ags_osc_client_read_bytes, 1680
ags_osc_client_resolve, 1679
ags_osc_client_set_flags, 1679
ags_osc_client_test_flags, 1678
ags_osc_client_unset_flags, 1679
ags_osc_client_write_bytes, 1680
AGS_OSC_CONFIG_CONTROLLER, 1724
ags_osc_config_controller_apply_config, 1723
AGS_OSC_CONFIG_CONTROLLER_CLASS, 1724
AGS_OSC_CONFIG_CONTROLLER_GET_CLASS, 1724
ags_osc_config_controller_get_type, 1724
ags_osc_config_controller_new, 1724
AGS_OSC_CONNECTION, 1703
AGS_OSC_CONNECTION_CHUNK_SIZE, 1704
AGS_OSC_CONNECTION_CLASS, 1703
ags_osc_connection_close, 1702
AGS_OSC_CONNECTION_DEAD_LINE_USEC, 1704
AGS_OSC_CONNECTION_DEFAULT_CACHE_DATA_LENGT 1704
AGS_OSC_CONNECTION_GET_CLASS, 1703
AGS_OSC_CONNECTION_GET_OBJ_MUTEX, 1700
ags_osc_connection_get_type, 1703
ags_osc_connection_new, 1702
ags_osc_connection_read_bytes, 1701
ags_osc_connection_set_flags, 1700
ags_osc_connection_test_flags, 1700
ags_osc_connection_timeout_expired, 1701
AGS_OSC_CONNECTION_TIMEOUT_USEC, 1703
ags_osc_connection_unset_flags, 1701
ags_osc_connection_write_response, 1702
AGS_OSC_CONTROLLER, 1717
AGS_OSC_CONTROLLER_CLASS, 1718
AGS_OSC_CONTROLLER_GET_CLASS, 1718
AGS_OSC_CONTROLLER_GET_OBJ_MUTEX, 1717
ags_osc_controller_get_type, 1718
ags_osc_controller_new, 1717
AGS_OSC_EXPORT_CONTROLLER, 1727
AGS_OSC_EXPORT_CONTROLLER_CLASS, 1727
ags_osc_export_controller_do_export, 1726
AGS_OSC_EXPORT_CONTROLLER_GET_CLASS, 1727
ags_osc_export_controller_get_type, 1728
ags_osc_export_controller_new, 1727
AGS_OSC_FRONT_CONTROLLER, 1733
ags_osc_front_controller_add_message, 1731
AGS_OSC_FRONT_CONTROLLER_CLASS, 1733
ags_osc_front_controller_do_request, 1732
ags_osc_front_controller_flags_get_type, 1733
AGS_OSC_FRONT_CONTROLLER_GET_CLASS, 1733
ags_osc_front_controller_get_type, 1733
ags_osc_front_controller_new, 1732
ags_osc_front_controller_remove_message, 1731
ags_osc_front_controller_set_flags, 1730
ags_osc_front_controller_start_delegate, 1731
ags_osc_front_controller_stop_delegate, 1732
ags_osc_front_controller_test_flags, 1730
ags_osc_front_controller_unset_flags, 1731
AGS_OSC_INFO_CONTROLLER, 1737
AGS_OSC_INFO_CONTROLLER_CLASS, 1737
AGS_OSC_INFO_CONTROLLER_GET_CLASS, 1737
ags_osc_info_controller_get_info, 1736
ags_osc_info_controller_get_type, 1737
ags_osc_info_controller_new, 1736
AGS_OSC_MESSAGE, 1708
AGS_OSC_MESSAGE_CLASS, 1708
AGS_OSC_MESSAGE_GET_CLASS, 1708
AGS_OSC_MESSAGE_GET_OBJ_MUTEX, 1707
ags_osc_message_get_type, 1708
ags_osc_message_new, 1708

ags_osc_message_sort_func, 1708
AGS_OSC_METER_CONTROLLER, 1745
ags_osc_meter_controller_add_monitor, 1743
AGS_OSC_METER_CONTROLLER_CLASS, 1745
ags_osc_meter_controller_contains_monitor, 1743
AGS_OSC_METER_CONTROLLER_DEFAULT_MONITOR_TIMEOUT, 1745
AGS_OSC_METER_CONTROLLER_GET_CLASS, 1745
ags_osc_meter_controller_get_type, 1745
AGS_OSC_METER_CONTROLLER_MONITOR, 1740
ags_osc_meter_controller_monitor_alloc, 1741
ags_osc_meter_controller_monitor_find_path, 1742
ags_osc_meter_controller_monitor_find_port, 1742
ags_osc_meter_controller_monitor_free, 1741
ags_osc_meter_controller_monitor_meter, 1744
ags_osc_meter_controller_monitor_ref, 1741
ags_osc_meter_controller_monitor_unref, 1742
ags_osc_meter_controller_new, 1744
ags_osc_meter_controller_remove_monitor, 1743
ags_osc_meter_controller_set_flags, 1740
ags_osc_meter_controller_start_monitor, 1743
ags_osc_meter_controller_stop_monitor, 1743
ags_osc_meter_controller_test_flags, 1740
ags_osc_meter_controller_unset_flags, 1740
AGS_OSC_NODE_CONTROLLER, 1749
AGS_OSC_NODE_CONTROLLER_CLASS, 1749
AGS_OSC_NODE_CONTROLLER_CONTEXT_PATH_LENGTH, 1749
AGS_OSC_NODE_CONTROLLER_GET_CLASS, 1749
ags_osc_node_controller_get_data, 1748
ags_osc_node_controller_get_type, 1749
ags_osc_node_controller_new, 1749
AGS_OSC_PARSER, 1671
ags_osc_parser_bundle, 1669
AGS_OSC_PARSER_CLASS, 1671
AGS_OSC_PARSER_GET_CLASS, 1671
AGS_OSC_PARSER_GET_OBJ_MUTEX, 1666
ags_osc_parser_get_type, 1671
AGS_OSC_PARSER_MAX_TEXT_LENGTH, 1671
ags_osc_parser_message, 1670
ags_osc_parser_new, 1670
ags_osc_parser_on_error, 1668
ags_osc_parser_osc_getc, 1667
ags_osc_parser_packet, 1669
ags_osc_parser_parse_bytes, 1668
ags_osc_parser_parse_full, 1668
ags_osc_parser_read_gdouble, 1667
ags_osc_parser_read_gfloat, 1666
ags_osc_parser_read_gint32, 1666
ags_osc_parser_read_gint64, 1666
ags_osc_parser_read_text, 1667
ags_osc_parser_value, 1670
AGS_OSC_PLUGIN_CONTROLLER, 1752
ags_osc_plugin_controller_do_request, 1751
AGS_OSC_PLUGIN_CONTROLLER_GET_INTERFACE, 1752
ags_osc_plugin_controller_get_type, 1752
AGS_OSC_PLUGIN_CONTROLLER_INTERFACE, 1752
AGS_OSC_RENEW_CONTROLLER, 1755
AGS_OSC_RENEW_CONTROLLER_CLASS, 1755
AGS_OSC_RENEW_CONTROLLER_CONTEXT_PATH_LENG 1755
AGS_OSC_RENEW_CONTROLLER_GET_CLASS, 1755
ags_osc_renew_controller_get_type, 1755
ags_osc_renew_controller_new, 1754
ags_osc_renew_controller_set_data, 1754
AGS_OSC_RESPONSE, 1713
AGS_OSC_RESPONSE_CLASS, 1713
AGS_OSC_RESPONSE_DEFAULT_CHUNK_SIZE, 1714
AGS_OSC_RESPONSE_ERROR_MESSAGE_CHUNK_SIZE_E 1714
AGS_OSC_RESPONSE_ERROR_MESSAGE_MALFORMED_R 1714
AGS_OSC_RESPONSE_ERROR_MESSAGE_MISSING_INDEX 1714
AGS_OSC_RESPONSE_ERROR_MESSAGE_SERVER_FAILUI 1714
AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_ARC 1714
AGS_OSC_RESPONSE_ERROR_MESSAGE_UNKNOWN_PAT 1714
AGS_OSC_RESPONSE_GET_CLASS, 1713
AGS_OSC_RESPONSE_GET_OBJ_MUTEX, 1712
ags_osc_response_get_type, 1713
ags_osc_response_new, 1713
ags_osc_response_set_flags, 1712
ags_osc_response_test_flags, 1712
ags_osc_response_unset_flags, 1712
AGS_OSC_SERVER, 1693
ags_osc_server_add_connection, 1690
ags_osc_server_add_controller, 1690
ags_osc_server_add_default_controller, 1691
AGS_OSC_SERVER_CLASS, 1693
AGS_OSC_SERVER_DEFAULT_BACKLOG, 1694
AGS_OSC_SERVER_DEFAULT_DOMAIN, 1693
AGS_OSC_SERVER_DEFAULT_INET4_ADDRESS, 1693
AGS_OSC_SERVER_DEFAULT_INET6_ADDRESS, 1694
AGS_OSC_SERVER_DEFAULT_MAX_ADDRESS_LENGTH, 1693
AGS_OSC_SERVER_DEFAULT_MAX_CONNECTIONS, 1694
AGS_OSC_SERVER_DEFAULT_SERVER_PORT, 1693
ags_osc_server_dispatch, 1692
ags_osc_server_flags_get_type, 1693
AGS_OSC_SERVER_GET_CLASS, 1693
AGS_OSC_SERVER_GET_OBJ_MUTEX, 1688
ags_osc_server_get_type, 1693
ags_osc_server_listen, 1692
ags_osc_server_new, 1692
ags_osc_server_remove_connection, 1690
ags_osc_server_remove_controller, 1690
ags_osc_server_set_flags, 1689
ags_osc_server_start, 1691
ags_osc_server_stop, 1691

ags_osc_server_test_flags, 1689
ags_osc_server_unset_flags, 1689
AGS_OSC_STATUS_CONTROLLER, 1758
AGS_OSC_STATUS_CONTROLLER_CLASS, 1758
AGS_OSC_STATUS_CONTROLLER_GET_CLASS, 1758
ags_osc_status_controller_get_status, 1757
ags_osc_status_controller_get_type, 1758
ags_osc_status_controller_new, 1758
AGS_OSC_UTIL_ADDRESS_PATTERN_VALID_CHARS_PATTERNS, 1651
ags_osc_util_get_type, 1649
ags_osc_util_meta_data, 1647
ags_osc_util_mime_header, 1647
AGS_OSC_UTIL_SLIP_CHUNK_LENGTH, 1651
ags_osc_util_slip_decode, 1648
ags_osc_util_slip_encode, 1648
AGS_OSC_UTIL_SLIP_END, 1651
AGS_OSC_UTIL_SLIP_ESC, 1651
AGS_OSC_UTIL_SLIP_ESC_END, 1651
AGS_OSC_UTIL_SLIP_ESC_ESC, 1651
ags_osc_util_timetag_now, 1649
AGS_OSC_UTIL_TYPE_TAG_STRING_ALL, 1651
AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_END, 1651
AGS_OSC_UTIL_TYPE_TAG_STRING_ARRAY_START, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_BLOB, 1649
AGS_OSC_UTIL_TYPE_TAG_STRING_CHAR, 1650
ags_osc_util_type_tag_string_count_type, 1647
AGS_OSC_UTIL_TYPE_TAG_STRING_DOUBLE, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_FALSE, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_FLOAT, 1649
AGS_OSC_UTIL_TYPE_TAG_STRING_INFINITE, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_INT32, 1649
AGS_OSC_UTIL_TYPE_TAG_STRING_INT64, 1649
AGS_OSC_UTIL_TYPE_TAG_STRING_MIDI, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_NIL, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_RGBA, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_STRING, 1649
AGS_OSC_UTIL_TYPE_TAG_STRING_SYMBOL, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_TIMETAG, 1650
AGS_OSC_UTIL_TYPE_TAG_STRING_TRUE, 1650
AGS_OSC_WEBSOCKET_CONNECTION, 1765
AGS_OSC_WEBSOCKET_CONNECTION_CLASS, 1765
ags_osc_websocket_connection_find_resource_id, 1764
AGS_OSC_WEBSOCKET_CONNECTION_GET_CLASS, 1766
ags_osc_websocket_connection_get_type, 1766
ags_osc_websocket_connection_new, 1765
AGS_OSC_XMLRPC_CONTROLLER, 1774
ags_osc_xmlrpc_controller_add_message, 1772
AGS_OSC_XMLRPC_CONTROLLER_CLASS, 1774
AGS_OSC_XMLRPC_CONTROLLER_DEFAULT_DELEGATE_TIMEOUT, 1774
ags_osc_xmlrpc_controller_flags_get_type, 1774
AGS_OSC_XMLRPC_CONTROLLER_GET_CLASS, 1774
ags_osc_xmlrpc_controller_get_type, 1774
ags_osc_xmlrpc_controller_new, 1773
ags_osc_xmlrpc_controller_remove_message, 1773
ags_osc_xmlrpc_controller_set_flags, 1772
ags_osc_xmlrpc_controller_start_delegate, 1773
ags_osc_xmlrpc_controller_stop_delegate, 1773
ags_osc_xmlrpc_controller_test_flags, 1771
ags_osc_xmlrpc_controller_unset_flags, 1772
AGS_OSC_XMLRPC_MESSAGE, 1769
AGS_OSC_XMLRPC_MESSAGE_CLASS, 1769
AGS_OSC_XMLRPC_MESSAGE_GET_CLASS, 1769
ags_osc_xmlrpc_message_get_type, 1769
ags_osc_xmlrpc_message_new, 1768
AGS_OSC_XMLRPC_SERVER, 1762
ags_osc_xmlrpc_server_add_default_controller, 1761
ags_osc_xmlrpc_server_add_websocket_handler, 1761
AGS_OSC_XMLRPC_SERVER_CLASS, 1762
AGS_OSC_XMLRPC_SERVER_DEFAULT_CHUNK_SIZE, 1762
AGS_OSC_XMLRPC_SERVER_GET_CLASS, 1762
ags_osc_xmlrpc_server_get_type, 1762
ags_osc_xmlrpc_server_new, 1762
AgsOscActionController, 1721
AgsOscActionController::run-action, 1722
AgsOscActionControllerClass, 1721
AgsOscBufferUtil, 1645
AgsOscBuilder, 1661
AgsOscBuilder::append-bundle, 1662
AgsOscBuilder::append-message, 1662
AgsOscBuilder::append-packet, 1663
AgsOscBuilder::append-value, 1663
AgsOscBuilder::on-error, 1664
AgsOscBuilder::osc-putc, 1664
AgsOscBuilderBundle, 1660
AgsOscBuilderClass, 1661
AgsOscBuilderFlags, 1660
AgsOscBuilderMessage, 1661
AgsOscBuilderPacket, 1660
AgsOscClient, 1683
AgsOscClient::connect, 1685
AgsOscClient::read-bytes, 1685
AgsOscClient::resolve, 1685
AgsOscClient::write-bytes, 1686
AgsOscClient:domain, 1684
AgsOscClient:ip4, 1684
AgsOscClient:ip6, 1684
AgsOscClient:server-port, 1684
AgsOscClientClass, 1683
AgsOscClientFlags, 1683
AgsOscConfigController, 1725
AgsOscConfigController::apply-config, 1725
AgsOscConfigControllerClass, 1725
AgsOscConnection, 1704
AgsOscConnection::close, 1705
AgsOscConnection::read-bytes, 1706
AgsOscConnection::write-response, 1706
AgsOscConnection:ip4, 1705
AgsOscConnection:ip6, 1705
AgsOscConnection:osc-server, 1705

AgsOscConnectionClass, 1704
AgsOscConnectionFlags, 1704
AgsOscController, 1718
AgsOscController:context-path, 1718
AgsOscController:osc-server, 1719
AgsOscControllerClass, 1718
AgsOscExportController, 1728
AgsOscExportController::do-export, 1728
AgsOscExportControllerClass, 1728
AgsOscFrontController, 1733
AgsOscFrontController::do-request, 1734
AgsOscFrontController::start-delegate, 1734
AgsOscFrontController::stop-delegate, 1735
AgsOscFrontControllerClass, 1734
AgsOscFrontControllerFlags, 1733
AgsOscInfoController, 1737
AgsOscInfoController::get-info, 1738
AgsOscInfoControllerClass, 1738
AgsOscMessage, 1709
AgsOscMessage:immediately, 1709
AgsOscMessage:message, 1709
AgsOscMessage:message-size, 1709
AgsOscMessage:osc-connection, 1710
AgsOscMessage:tv-fraction, 1710
AgsOscMessage:tv-sec, 1710
AgsOscMessageClass, 1709
AgsOscMeterController, 1746
AgsOscMeterController::monitor-meter, 1746
AgsOscMeterController::start-monitor, 1746
AgsOscMeterController::stop-monitor, 1747
AgsOscMeterControllerClass, 1746
AgsOscMeterControllerFlags, 1745
AgsOscMeterControllerMonitor, 1745
AgsOscNodeController, 1750
AgsOscNodeController::get-data, 1750
AgsOscNodeControllerClass, 1750
AgsOscParser, 1672
AgsOscParser::bundle, 1672
AgsOscParser::message, 1673
AgsOscParser::on-error, 1673
AgsOscParser::osc-getc, 1674
AgsOscParser::packet, 1674
AgsOscParser::parse-bytes, 1674
AgsOscParser::parse-full, 1675
AgsOscParser::value, 1675
AgsOscParserClass, 1672
AgsOscParserFlags, 1671
AgsOscPluginController, 1753
AgsOscPluginControllerInterface, 1753
AgsOscRenewController, 1755
AgsOscRenewController::set-data, 1756
AgsOscRenewControllerClass, 1755
AgsOscResponse, 1715
AgsOscResponse:error-message, 1715
AgsOscResponse:osc-message, 1715
AgsOscResponse:packet, 1715
AgsOscResponse:packet-size, 1716
AgsOscResponseClass, 1715
AgsOscResponseFlags, 1714
AgsOscServer, 1695
AgsOscServer::dispatch, 1697
AgsOscServer::listen, 1697
AgsOscServer::start, 1698
AgsOscServer::stop, 1698
AgsOscServer:connection, 1695
AgsOscServer:controller, 1696
AgsOscServer:domain, 1696
AgsOscServer:front-controller, 1696
AgsOscServer:ip4, 1696
AgsOscServer:ip6, 1697
AgsOscServer:server-port, 1697
AgsOscServerClass, 1695
AgsOscServerFlags, 1694
AgsOscStatusController, 1759
AgsOscStatusController::get-status, 1759
AgsOscStatusControllerClass, 1759
AgsOscUtil, 1651
AgsOscWebsocketConnection, 1766
AgsOscWebsocketConnection:login, 1766
AgsOscWebsocketConnection:resource-id, 1766
AgsOscWebsocketConnection:security-context, 1767
AgsOscWebsocketConnection:security-token, 1767
AgsOscWebsocketConnection:websocket-connection, 1767
AgsOscWebsocketConnectionClass, 1766
AgsOscXmlrpcController, 1775
AgsOscXmlrpcController::start-delegate, 1775
AgsOscXmlrpcController::stop-delegate, 1776
AgsOscXmlrpcController:osc-xmlrpc-server, 1775
AgsOscXmlrpcControllerClass, 1775
AgsOscXmlrpcControllerFlags, 1775
AgsOscXmlrpcMessage, 1769
AgsOscXmlrpcMessage:query, 1769
AgsOscXmlrpcMessage:server-msg, 1770
AgsOscXmlrpcMessageClass, 1769
AgsOscXmlrpcServer, 1763
AgsOscXmlrpcServer:osc-xmlrpc-controller, 1763
AgsOscXmlrpcServer:xmlrpc-server, 1763
AgsOscXmlrpcServerClass, 1763
AGS_OSS_DEVIN, 677
ags_oss_devin_adjust_delay_and_attack, 676
AGS_OSS_DEVIN_CLASS, 677
AGS_OSS_DEVIN_DEFAULT_APP_BUFFER_SIZE, 678
AGS_OSS_DEVIN_DEFAULT_BACKEND_BUFFER_SIZE, 678
AGS_OSS_DEVIN_DEFAULT_OSS_DEVICE, 678
AGS_OSS_DEVIN_DEFAULT_OSS_MIXER_DEVICE, 678
AGS_OSS_DEVIN_ERROR, 680
ags_oss_devin_error_quark, 675
ags_oss_devin_flags_get_type, 677
AGS_OSS_DEVIN_GET_CLASS, 677
AGS_OSS_DEVIN_GET_OBJ_MUTEX, 674
ags_oss_devin_get_type, 677
ags_oss_devin_new, 677
ags_oss_devin_realloc_buffer, 676

ags_oss_devin_set_flags, 675
ags_oss_devin_switch_buffer_flag, 676
ags_oss_devin_test_flags, 675
ags_oss_devin_unset_flags, 675
AGS_OSS_DEVOUT, 687
ags_oss_devout_adjust_delay_and_attack, 686
AGS_OSS_DEVOUT_CLASS, 687
AGS_OSS_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 688
AGS_OSS_DEVOUT_DEFAULT_BACKEND_BUFFER_SIZE, 688
AGS_OSS_DEVOUT_DEFAULT_OSS_DEVICE, 687
AGS_OSS_DEVOUT_DEFAULT_OSS_MIXER_DEVICE, 687
AGS_OSS_DEVOUT_ERROR, 689
ags_oss_devout_error_quark, 684
ags_oss_devout_flags_get_type, 687
AGS_OSS_DEVOUT_GET_CLASS, 687
AGS_OSS_DEVOUT_GET_OBJ_MUTEX, 684
ags_oss_devout_get_type, 687
ags_oss_devout_new, 686
ags_oss_devout_realloc_buffer, 686
ags_oss_devout_set_flags, 685
ags_oss_devout_switch_buffer_flag, 686
ags_oss_devout_test_flags, 685
ags_oss_devout_unset_flags, 685
AGS_OSS_MIDIIN, 696
AGS_OSS_MIDIIN_CLASS, 696
AGS_OSS_MIDIIN_DEFAULT_APP_BUFFER_SIZE, 697
AGS_OSS_MIDIIN_DEFAULT_BACKEND_BUFFER_SIZE, 697
AGS_OSS_MIDIIN_DEFAULT_BUFFER_SIZE, 697
AGS_OSS_MIDIIN_DEFAULT_OSS_DEVICE, 696
AGS_OSS_MIDIIN_DEFAULT_OSS_MIXER_DEVICE, 697
AGS_OSS_MIDIIN_ERROR, 699
ags_oss_midiin_error_quark, 694
ags_oss_midiin_flags_get_type, 696
AGS_OSS_MIDIIN_GET_CLASS, 696
AGS_OSS_MIDIIN_GET_OBJ_MUTEX, 694
ags_oss_midiin_get_type, 696
ags_oss_midiin_new, 695
ags_oss_midiin_set_flags, 695
ags_oss_midiin_switch_buffer_flag, 695
ags_oss_midiin_test_flags, 694
ags_oss_midiin_unset_flags, 695
AgsOssDevin, 680
AgsOssDevin:attack, 680
AgsOssDevin:bpm, 681
AgsOssDevin:buffer, 681
AgsOssDevin:buffer-size, 681
AgsOssDevin:delay-factor, 681
AgsOssDevin:device, 682
AgsOssDevin:dsp-channels, 682
AgsOssDevin:format, 682
AgsOssDevin:pcm-channels, 682
AgsOssDevin:samplerate, 683
AgsOssDevinAppBufferMode, 679
AgsOssDevinBackendBufferMode, 679
AgsOssDevinClass, 680
AgsOssDevinError, 680
AgsOssDevinFlags, 678
AgsOssDevout, 690
AgsOssDevout:attack, 690
AgsOssDevout:bpm, 690
AgsOssDevout:buffer, 691
AgsOssDevout:buffer-size, 691
AgsOssDevout:delay-factor, 691
AgsOssDevout:device, 691
AgsOssDevout:dsp-channels, 692
AgsOssDevout:format, 692
AgsOssDevout:pcm-channels, 692
AgsOssDevout:samplerate, 692
AgsOssDevoutAppBufferMode, 688
AgsOssDevoutBackendBufferMode, 689
AgsOssDevoutClass, 690
AgsOssDevoutError, 690
AgsOssDevoutFlags, 688
AgsOssMidiin, 700
AgsOssMidiin:attack, 700
AgsOssMidiin:bpm, 700
AgsOssMidiin:buffer, 701
AgsOssMidiin:delay-factor, 701
AgsOssMidiin:device, 701
AgsOssMidiinAppBufferMode, 698
AgsOssMidiinBackendBufferMode, 698
AgsOssMidiinClass, 700
AgsOssMidiinError, 700
AgsOssMidiinFlags, 697
AgsOssMidiinSyncFlags, 699
AGS_OUTPUT, 1099
AgsOutput, 1099
AGS_OUTPUT_CLASS, 1099
ags_output_find_first_input_recycling, 1097
ags_output_find_last_input_recycling, 1098
AGS_OUTPUT_GET_CLASS, 1099
ags_output_get_type, 1099
ags_output_new, 1098
AgsOutputClass, 1099

P

AGS_PATTERN, 1299
AgsPattern, 1301
AgsPattern:channel, 1301
AgsPattern:current-bit, 1301
AgsPattern:first-index, 1302
AgsPattern:offset, 1302
AgsPattern:port, 1302
AgsPattern:second-index, 1302
AgsPattern:timestamp, 1303
AGS_PATTERN_CLASS, 1300
AGS_PATTERN_DEFAULT_BPM, 1300
AGS_PATTERN_DEFAULT_DURATION, 1301
AGS_PATTERN_DEFAULT_JIFFIE, 1300
AGS_PATTERN_DEFAULT_LENGTH, 1300
AGS_PATTERN_DEFAULT_OFFSET, 1301

ags_pattern_find_near_timestamp, 1295
ags_pattern_get_bit, 1298
ags_pattern_get_channel, 1295
AGS_PATTERN_GET_CLASS, 1300
ags_pattern_get_dim, 1297
ags_pattern_get_note, 1299
AGS_PATTERN_GET_OBJ_MUTEX, 1293
ags_pattern_get_obj_mutex, 1294
ags_pattern_get_port, 1297
ags_pattern_get_timestamp, 1296
ags_pattern_get_type, 1300
ags_pattern_is_empty, 1298
AGS_PATTERN_MAXIMUM_NOTE_LENGTH, 1300
AGS_PATTERN_MINIMUM_NOTE_LENGTH, 1300
ags_pattern_new, 1299
ags_pattern_set_channel, 1296
ags_pattern_set_dim, 1297
ags_pattern_set_flags, 1294
ags_pattern_set_port, 1297
ags_pattern_set_timestamp, 1296
ags_pattern_test_flags, 1294
AGS_PATTERN_TICS_PER_BEAT, 1300
ags_pattern_toggle_bit, 1298
ags_pattern_unset_flags, 1295
AgsPatternClass, 1301
AGS_PEAK_UTIL, 351
ags_peak_util_alloc, 343
ags_peak_util_compute, 351
ags_peak_util_compute_complex, 350
ags_peak_util_compute_double, 350
ags_peak_util_compute_float, 350
ags_peak_util_compute_s16, 349
ags_peak_util_compute_s24, 349
ags_peak_util_compute_s32, 349
ags_peak_util_compute_s64, 350
ags_peak_util_compute_s8, 349
ags_peak_util_copy, 343
AGS_PEAK_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT, 351
AGS_PEAK_UTIL_DEFAULT_FORMAT, 351
AGS_PEAK_UTIL_DEFAULT_HARMONIC_RATE, 351
AGS_PEAK_UTIL_DEFAULT_PRESSURE_FACTOR, 352
AGS_PEAK_UTIL_DEFAULT_SAMPLERATE, 351
ags_peak_util_free, 343
ags_peak_util_get_buffer_length, 345
ags_peak_util_get_format, 345
ags_peak_util_get_harmonic_rate, 347
ags_peak_util_get_peak, 348
ags_peak_util_get_pressure_factor, 347
ags_peak_util_get_samplerate, 346
ags_peak_util_get_source, 344
ags_peak_util_get_source_stride, 344
ags_peak_util_get_type, 351
ags_peak_util_set_buffer_length, 345
ags_peak_util_set_format, 346
ags_peak_util_set_harmonic_rate, 347
ags_peak_util_set_peak, 348
ags_peak_util_set_pressure_factor, 348
ags_peak_util_set_samplerate, 346
ags_peak_util_set_source, 344
ags_peak_util_set_source_stride, 345
AgsPeakUtil, 352
ags_phase_shift_util_alloc, 247
ags_phase_shift_util_copy, 247
AGS_PHASE_SHIFT_UTIL_DEFAULT_FORMAT, 256
ags_phase_shift_util_free, 247
ags_phase_shift_util_get_amount, 252
ags_phase_shift_util_get_buffer_length, 250
ags_phase_shift_util_get_destination, 248
ags_phase_shift_util_get_destination_stride, 248
ags_phase_shift_util_get_format, 251
ags_phase_shift_util_get_frequency, 252
ags_phase_shift_util_get_phase, 253
ags_phase_shift_util_get_source, 249
ags_phase_shift_util_get_source_stride, 250
ags_phase_shift_util_get_type, 256
ags_phase_shift_util_process, 255
ags_phase_shift_util_process_complex, 255
ags_phase_shift_util_process_double, 255
ags_phase_shift_util_process_float, 255
ags_phase_shift_util_process_s16, 254
ags_phase_shift_util_process_s24, 254
ags_phase_shift_util_process_s32, 254
ags_phase_shift_util_process_s64, 254
ags_phase_shift_util_process_s8, 253
ags_phase_shift_util_set_amount, 253
ags_phase_shift_util_set_buffer_length, 251
ags_phase_shift_util_set_destination, 248
ags_phase_shift_util_set_destination_stride, 249
ags_phase_shift_util_set_format, 251
ags_phase_shift_util_set_frequency, 252
ags_phase_shift_util_set_phase, 253
ags_phase_shift_util_set_source, 249
ags_phase_shift_util_set_source_stride, 250
AgsPhaseShiftUtil, 256
AGS_PLAYBACK, 1333
AgsPlayback, 1334
AgsPlayback:audio-channel, 1334
AgsPlayback:channel, 1334
AgsPlayback:play-note, 1335
AgsPlayback:playback-domain, 1335
AGS_PLAYBACK_CLASS, 1333
AGS_PLAYBACK_DOMAIN, 1326
ags_playback_domain_add_playback, 1324
AGS_PLAYBACK_DOMAIN_CLASS, 1326
ags_playback_domain_get_audio_thread, 1324
AGS_PLAYBACK_DOMAIN_GET_CLASS, 1326
AGS_PLAYBACK_DOMAIN_GET_OBJ_MUTEX, 1323
ags_playback_domain_get_type, 1326
ags_playback_domain_insert_playback, 1325
ags_playback_domain_new, 1326
ags_playback_domain_remove_playback, 1325
ags_playback_domain_set_audio_thread, 1324
ags_playback_domain_set_flags, 1323

ags_playback_domain_test_flags, 1323
ags_playback_domain_unset_flags, 1323
ags_playback_find_channel, 1332
ags_playback_flags_get_type, 1333
ags_playback_get_channel_thread, 1331
AGS_PLAYBACK_GET_CLASS, 1333
AGS_PLAYBACK_GET_OBJ_MUTEX, 1329
ags_playback_get_recall_id, 1331
ags_playback_get_type, 1333
ags_playback_new, 1332
ags_playback_set_channel_thread, 1330
ags_playback_set_flags, 1330
ags_playback_set_recall_id, 1331
ags_playback_test_flags, 1329
ags_playback_unset_flags, 1330
AgsPlaybackClass, 1334
AgsPlaybackDomain, 1327
AgsPlaybackDomain:audio, 1327
AgsPlaybackDomain:input-playback, 1327
AgsPlaybackDomain:output-playback, 1328
AgsPlaybackDomainClass, 1327
AgsPlaybackDomainFlags, 1327
AgsPlaybackFlags, 1333
AGS_PLUGIN_NAME_VOLUME, 25
AGS_PLUGIN_PORT, 35
AGS_PLUGIN_PORT_CLASS, 35
ags_plugin_port_find_port_index, 34
ags_plugin_port_find_symbol, 33
ags_plugin_port_flags_get_type, 35
AGS_PLUGIN_PORT_GET_CLASS, 35
ags_plugin_port_get_default_value, 33
ags_plugin_port_get_lower_value, 32
AGS_PLUGIN_PORT_GET_OBJ_MUTEX, 26
ags_plugin_port_get_obj_mutex, 27
ags_plugin_port_get_port_index, 28
ags_plugin_port_get_port_name, 28
ags_plugin_port_get_port_symbol, 29
ags_plugin_port_get_scale_point, 30
ags_plugin_port_get_scale_steps, 30
ags_plugin_port_get_scale_value, 31
ags_plugin_port_get_type, 35
ags_plugin_port_get_upper_value, 32
ags_plugin_port_new, 34
ags_plugin_port_set_default_value, 33
ags_plugin_port_set_flags, 27
ags_plugin_port_set_lower_value, 32
ags_plugin_port_set_port_index, 28
ags_plugin_port_set_port_name, 29
ags_plugin_port_set_port_symbol, 29
ags_plugin_port_set_scale_point, 31
ags_plugin_port_set_scale_steps, 30
ags_plugin_port_set_scale_value, 31
ags_plugin_port_set_upper_value, 33
ags_plugin_port_test_flags, 27
ags_plugin_port_unset_flags, 28
AgsPluginPort, 36
AgsPluginPort:default-value, 36
AgsPluginPort:lower-value, 37
AgsPluginPort:port-index, 37
AgsPluginPort:port-name, 37
AgsPluginPort:port-symbol, 37
AgsPluginPort:scale-point, 37
AgsPluginPort:scale-steps, 38
AgsPluginPort:scale-value, 38
AgsPluginPort:upper-value, 38
AgsPluginPortClass, 36
AgsPluginPortFlags, 35
AGS_PORT, 1157
AgsPort, 1158
AgsPort::safe-get-property, 1160
AgsPort::safe-read, 1161
AgsPort::safe-set-property, 1161
AgsPort::safe-write, 1161
AgsPort:automation, 1158
AgsPort:control-port, 1158
AgsPort:conversion, 1158
AgsPort:plugin-name, 1159
AgsPort:plugin-port, 1159
AgsPort:port-value-is-pointer, 1159
AgsPort:port-value-length, 1159
AgsPort:port-value-size, 1159
AgsPort:port-value-type, 1160
AgsPort:specifier, 1160
ags_port_add_automation, 1156
AGS_PORT_CLASS, 1157
ags_port_find_plugin_port, 1155
ags_port_find_specifier, 1155
ags_port_flags_get_type, 1157
AGS_PORT_GET_CLASS, 1157
AGS_PORT_GET_OBJ_MUTEX, 1152
ags_port_get_type, 1157
ags_port_new, 1156
ags_port_remove_automation, 1156
ags_port_safe_get_property, 1154
ags_port_safe_read, 1153
ags_port_safe_read_raw, 1153
ags_port_safe_set_property, 1154
ags_port_safe_write, 1154
ags_port_safe_write_raw, 1154
ags_port_set_flags, 1153
ags_port_test_flags, 1152
ags_port_unset_flags, 1153
AgsPortClass, 1158
AgsPortFlags, 1157
AGS_PRESET, 1314
AgsPreset, 1315
AgsPreset:audio, 1315
AgsPreset:audio-channel-end, 1315
AgsPreset:audio-channel-start, 1315
AgsPreset:pad-end, 1316
AgsPreset:pad-start, 1316
AgsPreset:preset-name, 1316
AgsPreset:scope, 1316
AgsPreset:x-end, 1316

AgsPreset:x-start, 1317
ags_preset_add_parameter, 1312
AGS_PRESET_CLASS, 1314
AGS_PRESET_ERROR, 1314
ags_preset_error_quark, 1304
ags_preset_find_name, 1312
ags_preset_find_scope, 1312
ags_preset_get_audio, 1306
ags_preset_get_audio_channel_end, 1308
ags_preset_get_audio_channel_start, 1308
AGS_PRESET_GET_CLASS, 1314
AGS_PRESET_GET_OBJ_MUTEX, 1304
ags_preset_get_obj_mutex, 1305
ags_preset_get_pad_end, 1310
ags_preset_get_pad_start, 1309
ags_preset_get_parameter, 1313
ags_preset_get_preset_name, 1307
ags_preset_get_scope, 1307
ags_preset_get_type, 1314
ags_preset_get_x_end, 1311
ags_preset_get_x_start, 1310
ags_preset_new, 1313
ags_preset_remove_parameter, 1313
ags_preset_set_audio, 1306
ags_preset_set_audio_channel_end, 1309
ags_preset_set_audio_channel_start, 1308
ags_preset_set_flags, 1305
ags_preset_set_pad_end, 1310
ags_preset_set_pad_start, 1309
ags_preset_set_preset_name, 1308
ags_preset_set_scope, 1307
ags_preset_set_x_end, 1311
ags_preset_set_x_start, 1311
ags_preset_test_flags, 1305
ags_preset_unset_flags, 1306
AgsPresetClass, 1315
AgsPresetError, 1314
AGS_PULSE_CLIENT, 783
ags_pulse_client_activate, 781
ags_pulse_client_add_device, 780
ags_pulse_client_add_port, 781
AGS_PULSE_CLIENT_CLASS, 783
ags_pulse_client_close, 780
ags_pulse_client_deactivate, 782
ags_pulse_client_find, 779
ags_pulse_client_find_uuid, 779
ags_pulse_client_flags_get_type, 783
AGS_PULSE_CLIENT_GET_CLASS, 783
AGS_PULSE_CLIENT_GET_OBJ_MUTEX, 778
ags_pulse_client_get_type, 783
ags_pulse_client_new, 782
ags_pulse_client_open, 780
ags_pulse_client_remove_device, 781
ags_pulse_client_remove_port, 781
ags_pulse_client_set_flags, 778
ags_pulse_client_test_flags, 778
ags_pulse_client_unset_flags, 779
AGS_PULSE_DEVIN, 796
ags_pulse_devin_adjust_delay_and_attack, 795
AGS_PULSE_DEVIN_CLASS, 796
AGS_PULSE_DEVIN_DEFAULT_APP_BUFFER_SIZE, 797
AGS_PULSE_DEVIN_ERROR, 799
ags_pulse_devin_error_quark, 794
ags_pulse_devin_flags_get_type, 797
AGS_PULSE_DEVIN_GET_CLASS, 796
AGS_PULSE_DEVIN_GET_OBJ_MUTEX, 794
ags_pulse_devin_get_type, 797
ags_pulse_devin_new, 796
ags_pulse_devin_realloc_buffer, 796
ags_pulse_devin_set_flags, 794
ags_pulse_devin_switch_buffer_flag, 795
ags_pulse_devin_test_flags, 794
ags_pulse_devin_unset_flags, 795
AGS_PULSE_DEVOUT, 806
ags_pulse_devout_adjust_delay_and_attack, 805
AGS_PULSE_DEVOUT_CLASS, 807
AGS_PULSE_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 807
AGS_PULSE_DEVOUT_ERROR, 809
ags_pulse_devout_error_quark, 804
ags_pulse_devout_flags_get_type, 807
AGS_PULSE_DEVOUT_GET_CLASS, 807
AGS_PULSE_DEVOUT_GET_OBJ_MUTEX, 804
ags_pulse_devout_get_type, 807
ags_pulse_devout_new, 806
ags_pulse_devout_realloc_buffer, 806
ags_pulse_devout_set_flags, 805
ags_pulse_devout_switch_buffer_flag, 805
ags_pulse_devout_test_flags, 804
ags_pulse_devout_unset_flags, 805
AGS_PULSE_PORT, 790
AGS_PULSE_PORT_CLASS, 790
AGS_PULSE_PORT_DEFAULT_CACHE_BUFFER_SIZE, 790
AGS_PULSE_PORT_DEFAULT_CACHE_COUNT, 790
ags_pulse_port_find, 787
ags_pulse_port_flags_get_type, 790
AGS_PULSE_PORT_GET_CLASS, 790
ags_pulse_port_get_fixed_size, 788
ags_pulse_port_get_latency, 789
AGS_PULSE_PORT_GET_OBJ_MUTEX, 786
ags_pulse_port_get_type, 790
ags_pulse_port_new, 789
ags_pulse_port_register, 788
ags_pulse_port_set_buffer_size, 789
ags_pulse_port_set_cache_buffer_size, 789
ags_pulse_port_set_flags, 787
ags_pulse_port_set_format, 789
ags_pulse_port_set_pcm_channels, 788
ags_pulse_port_set_samplerate, 788
ags_pulse_port_test_flags, 786
ags_pulse_port_unregister, 788
ags_pulse_port_unset_flags, 787
AGS_PULSE_SERVER, 775

ags_pulse_server_add_client, 773
AGS_PULSE_SERVER_CLASS, 775
ags_pulse_server_connect_client, 774
ags_pulse_server_disconnect_client, 774
ags_pulse_server_find_client, 772
ags_pulse_server_find_port, 773
ags_pulse_server_find_url, 772
AGS_PULSE_SERVER_GET_CLASS, 775
AGS_PULSE_SERVER_GET_OBJ_MUTEX, 771
ags_pulse_server_get_type, 775
ags_pulse_server_new, 774
ags_pulse_server_remove_client, 773
ags_pulse_server_set_flags, 771
ags_pulse_server_start_poll, 774
ags_pulse_server_test_flags, 771
ags_pulse_server_unset_flags, 771
AgsPulseClient, 784
AgsPulseClient:client-name, 784
AgsPulseClient:device, 784
AgsPulseClient:port, 784
AgsPulseClient:pulse-server, 785
AgsPulseClientClass, 784
AgsPulseClientFlags, 783
AgsPulseDevin, 799
AgsPulseDevin:attack, 800
AgsPulseDevin:bpm, 800
AgsPulseDevin:buffer, 800
AgsPulseDevin:buffer-size, 800
AgsPulseDevin:delay-factor, 801
AgsPulseDevin:device, 801
AgsPulseDevin:dsp-channels, 801
AgsPulseDevin:format, 801
AgsPulseDevin:pcm-channels, 802
AgsPulseDevin:pulse-client, 802
AgsPulseDevin:pulse-port, 802
AgsPulseDevin:samplerate, 802
AgsPulseDevinAppBufferMode, 798
AgsPulseDevinClass, 800
AgsPulseDevinError, 799
AgsPulseDevinFlags, 797
AgsPulseDevinSyncFlags, 798
AgsPulseDevout, 810
AgsPulseDevout:attack, 810
AgsPulseDevout:bpm, 810
AgsPulseDevout:buffer, 810
AgsPulseDevout:buffer-size, 811
AgsPulseDevout:delay-factor, 811
AgsPulseDevout:device, 811
AgsPulseDevout:dsp-channels, 811
AgsPulseDevout:format, 812
AgsPulseDevout:pcm-channels, 812
AgsPulseDevout:pulse-client, 812
AgsPulseDevout:pulse-port, 812
AgsPulseDevout:samplerate, 812
AgsPulseDevoutAppBufferMode, 808
AgsPulseDevoutClass, 810
AgsPulseDevoutError, 809
AgsPulseDevoutFlags, 807
AgsPulseDevoutSyncFlags, 808
AgsPulsePort, 791
AgsPulsePort:port-name, 792
AgsPulsePort:pulse-client, 792
AgsPulsePort:pulse-devin, 792
AgsPulsePort:pulse-devout, 792
AgsPulsePortClass, 791
AgsPulsePortFlags, 791
AgsPulseServer, 775
AgsPulseServer:default-pulse-client, 776
AgsPulseServer:default-soundcard, 776
AgsPulseServer:pulse-client, 776
AgsPulseServer:url, 776
AgsPulseServerClass, 775

R

AGS_RECALL, 1401
AgsRecall, 1404
AgsRecall::automate, 1409
AgsRecall::cancel, 1410
AgsRecall::check-rt-data, 1410
AgsRecall::child-added, 1410
AgsRecall::do-feedback, 1411
AgsRecall::done, 1411
AgsRecall::duplicate, 1412
AgsRecall::feed-input-queue, 1412
AgsRecall::feed-output-queue, 1413
AgsRecall::notify-dependency, 1413
AgsRecall::resolve-dependency, 1413
AgsRecall::run-init-inter, 1414
AgsRecall::run-init-post, 1414
AgsRecall::run-init-pre, 1414
AgsRecall::run-inter, 1415
AgsRecall::run-post, 1415
AgsRecall::run-pre, 1415
AgsRecall::stop-persistent, 1416
AgsRecall:audio-channel, 1405
AgsRecall:automation-port, 1405
AgsRecall:buffer-size, 1405
AgsRecall:child, 1406
AgsRecall:child-type, 1406
AgsRecall:effect, 1406
AgsRecall:effect-index, 1406
AgsRecall:filename, 1406
AgsRecall:format, 1407
AgsRecall:input-soundcard, 1407
AgsRecall:input-soundcard-channel, 1407
AgsRecall:line, 1407
AgsRecall:output-soundcard, 1407
AgsRecall:output-soundcard-channel, 1408
AgsRecall:pad, 1408
AgsRecall:parent, 1408
AgsRecall:port, 1408
AgsRecall:recall-container, 1408
AgsRecall:recall-dependency, 1409
AgsRecall:recall-id, 1409

AgsRecall:samplerate, 1409
ags_recall_add_child, 1384
ags_recall_add_port, 1383
ags_recall_add_recall_dependency, 1382
ags_recall_add_recall_handler, 1386
AGS_RECALL_AUDIO, 1419
AGS_RECALL_AUDIO_CLASS, 1419
ags_recall_audio_get_audio, 1417
AGS_RECALL_AUDIO_GET_CLASS, 1419
ags_recall_audio_get_type, 1419
ags_recall_audio_new, 1418
AGS_RECALL_AUDIO_RUN, 1423
AGS_RECALL_AUDIO_RUN_CLASS, 1423
ags_recall_audio_run_get_audio, 1421
AGS_RECALL_AUDIO_RUN_GET_CLASS, 1423
ags_recall_audio_run_get_recall_audio, 1422
ags_recall_audio_run_get_type, 1423
ags_recall_audio_run_new, 1423
ags_recall_audio_run_set_audio, 1422
ags_recall_audio_run_set_recall_audio, 1422
ags_recall_audio_set_audio, 1418
AGS_RECALL_AUDIO_SIGNAL, 1442
AGS_RECALL_AUDIO_SIGNAL_CLASS, 1442
AGS_RECALL_AUDIO_SIGNAL_GET_CLASS, 1442
ags_recall_audio_signal_get_type, 1442
ags_recall_audio_signal_new, 1442
ags_recall_automate, 1391
ags_recall_cancel, 1393
AGS_RECALL_CHANNEL, 1428
AGS_RECALL_CHANNEL_CLASS, 1428
ags_recall_channel_find_channel, 1427
AGS_RECALL_CHANNEL_GET_CLASS, 1428
ags_recall_channel_get_destination, 1426
ags_recall_channel_get_source, 1426
ags_recall_channel_get_type, 1428
ags_recall_channel_new, 1427
AGS_RECALL_CHANNEL_RUN, 1435
AGS_RECALL_CHANNEL_RUN_CLASS, 1435
AGS_RECALL_CHANNEL_RUN_GET_CLASS, 1435
ags_recall_channel_run_get_destination, 1433
ags_recall_channel_run_get_recall_audio, 1431
ags_recall_channel_run_get_recall_audio_run, 1431
ags_recall_channel_run_get_recall_channel, 1432
ags_recall_channel_run_get_source, 1433
ags_recall_channel_run_get_type, 1435
ags_recall_channel_run_new, 1434
ags_recall_channel_run_set_destination, 1433
ags_recall_channel_run_set_recall_audio, 1431
ags_recall_channel_run_set_recall_audio_run, 1432
ags_recall_channel_run_set_recall_channel, 1432
ags_recall_channel_run_set_source, 1434
ags_recall_channel_set_destination, 1426
ags_recall_channel_set_source, 1427
ags_recall_check_ability_flags, 1372
ags_recall_check_behaviour_flags, 1374
ags_recall_check_rt_data, 1390
ags_recall_check_sound_scope, 1375
ags_recall_check_staging_flags, 1376
ags_recall_check_state_flags, 1378
ags_recall_child_added, 1394
AGS_RECALL_CLASS, 1401
AGS_RECALL_CONTAINER, 1357
ags_recall_container_add, 1354
AGS_RECALL_CONTAINER_CLASS, 1357
ags_recall_container_find, 1356
ags_recall_container_flags_get_type, 1357
AGS_RECALL_CONTAINER_GET_CLASS, 1357
AGS_RECALL_CONTAINER_GET_OBJ_MUTEX, 1353
ags_recall_container_get_recall_audio, 1354
ags_recall_container_get_recall_audio_run, 1355
ags_recall_container_get_recall_channel, 1355
ags_recall_container_get_recall_channel_run, 1355
ags_recall_container_get_type, 1357
ags_recall_container_new, 1356
ags_recall_container_remove, 1354
ags_recall_container_set_flags, 1353
ags_recall_container_test_flags, 1353
ags_recall_container_unset_flags, 1353
AGS_RECALL_DEFAULT_BUILD_ID, 1402
AGS_RECALL_DEFAULT_VERSION, 1401
AGS_RECALL_DEPENDENCY, 1363
AGS_RECALL_DEPENDENCY_CLASS, 1363
ags_recall_dependency_find_dependency, 1361
ags_recall_dependency_find_dependency_by_provider, 1361
AGS_RECALL_DEPENDENCY_GET_CLASS, 1363
AGS_RECALL_DEPENDENCY_GET_OBJ_MUTEX, 1361
ags_recall_dependency_get_type, 1363
ags_recall_dependency_new, 1362
ags_recall_dependency_resolve, 1362
ags_recall_do_feedback, 1392
ags_recall_done, 1393
ags_recall_duplicate, 1394
ags_recall_feed_input_queue, 1391
ags_recall_feed_output_queue, 1392
ags_recall_find_provider, 1399
ags_recall_find_provider_with_recycling_context, 1399
ags_recall_find_recall_id_with_effect, 1396
ags_recall_find_recycling_context, 1398
ags_recall_find_template, 1396
ags_recall_find_type, 1396
ags_recall_find_type_with_recycling_context, 1398
ags_recall_flags_get_type, 1401
ags_recall_get_buffer_size, 1388
ags_recall_get_by_effect, 1395
ags_recall_get_children, 1384
AGS_RECALL_GET_CLASS, 1401
ags_recall_get_effect, 1379
ags_recall_get_effect_index, 1379
ags_recall_get_filename, 1378
ags_recall_get_format, 1389
ags_recall_get_input_soundcard, 1387
AGS_RECALL_GET_OBJ_MUTEX, 1368
ags_recall_get_obj_mutex, 1370
ags_recall_get_output_soundcard, 1386

ags_recall_get_port, 1382
ags_recall_get_recall_container, 1380
ags_recall_get_recall_dependency, 1381
ags_recall_get_recall_id, 1381
ags_recall_get_samplerate, 1388
ags_recall_get_sound_scope, 1375
ags_recall_get_type, 1401
ags_recall_global_get_children_lock_free, 1369
ags_recall_global_get_omit_event, 1369
ags_recall_global_get_performance_mode, 1369
ags_recall_global_get_rt_safe, 1370
ags_recall_global_set_omit_event, 1369
AGS_RECALL_HANDLER, 1368
ags_recall_handler_alloc, 1385
ags_recall_handler_free, 1385
AGS_RECALL_ID, 1341
ags_recall_id_check_sound_scope, 1336
ags_recall_id_check_staging_flags, 1338
ags_recall_id_check_state_flags, 1339
AGS_RECALL_ID_CLASS, 1342
ags_recall_id_find_parent_recycling_context, 1341
ags_recall_id_find_recycling_context, 1340
AGS_RECALL_ID_GET_CLASS, 1342
AGS_RECALL_ID_GET_OBJ_MUTEX, 1336
ags_recall_id_get_recycling_context, 1339
ags_recall_id_get_type, 1342
ags_recall_id_new, 1341
ags_recall_id_set_recycling_context, 1340
ags_recall_id_set_sound_scope, 1336
ags_recall_id_set_staging_flags, 1337
ags_recall_id_set_state_flags, 1338
ags_recall_id_test_staging_flags, 1337
ags_recall_id_test_state_flags, 1338
ags_recall_id_unset_staging_flags, 1337
ags_recall_id_unset_state_flags, 1339
ags_recall_is_done, 1395
ags_recall_lock_port, 1400
ags_recall_match_ability_flags_to_scope, 1373
ags_recall_new, 1400
ags_recall_notify_dependency, 1394
ags_recall_notify_dependency_mode_get_type, 1401
AGS_RECALL_RECYCLING, 1438
AGS_RECALL_RECYCLING_CLASS, 1438
AGS_RECALL_RECYCLING_GET_CLASS, 1438
ags_recall_recycling_get_type, 1438
ags_recall_recycling_new, 1438
ags_recall_remove_child, 1385
ags_recall_remove_port, 1383
ags_recall_remove_recall_dependency, 1382
ags_recall_remove_recall_handler, 1386
ags_recall_resolve_dependency, 1390
ags_recall_run_init_inter, 1390
ags_recall_run_init_post, 1391
ags_recall_run_init_pre, 1390
ags_recall_run_inter, 1392
ags_recall_run_post, 1392
ags_recall_run_pre, 1391
ags_recall_set_ability_flags, 1372
ags_recall_set_behaviour_flags, 1373
ags_recall_set_buffer_size, 1389
ags_recall_set_children, 1384
ags_recall_set_effect, 1379
ags_recall_set_effect_index, 1380
ags_recall_set_filename, 1378
ags_recall_set_flags, 1371
ags_recall_set_format, 1389
ags_recall_set_input_soundcard, 1387
ags_recall_set_output_soundcard, 1387
ags_recall_set_port, 1383
ags_recall_set_recall_container, 1380
ags_recall_set_recall_dependency, 1382
ags_recall_set_recall_id, 1381
ags_recall_set_samplerate, 1388
ags_recall_set_sound_scope, 1374
ags_recall_set_staging_flags, 1376
ags_recall_set_state_flags, 1377
ags_recall_stop_persistent, 1393
ags_recall_template_find_all_type, 1397
ags_recall_template_find_provider, 1399
ags_recall_template_find_type, 1397
ags_recall_test_ability_flags, 1371
ags_recall_test_behaviour_flags, 1373
ags_recall_test_flags, 1370
ags_recall_test_staging_flags, 1375
ags_recall_test_state_flags, 1377
ags_recall_unlock_port, 1400
ags_recall_unset_ability_flags, 1372
ags_recall_unset_behaviour_flags, 1374
ags_recall_unset_flags, 1371
ags_recall_unset_staging_flags, 1376
ags_recall_unset_state_flags, 1377
AgsRecallAudio, 1419
AgsRecallAudio:audio, 1420
AgsRecallAudioClass, 1419
AgsRecallAudioRun, 1424
AgsRecallAudioRun:audio, 1424
AgsRecallAudioRun:recall-audio, 1424
AgsRecallAudioRunClass, 1424
AgsRecallAudioSignal, 1443
AgsRecallAudioSignal:destination, 1443
AgsRecallAudioSignal:source, 1443
AgsRecallAudioSignalClass, 1443
AgsRecallChannel, 1428
AgsRecallChannel:destination, 1429
AgsRecallChannel:recall-audio, 1429
AgsRecallChannel:source, 1429
AgsRecallChannelClass, 1428
AgsRecallChannelRun, 1435
AgsRecallChannelRun:destination, 1435
AgsRecallChannelRun:recall-audio, 1436
AgsRecallChannelRun:recall-audio-run, 1436
AgsRecallChannelRun:recall-channel, 1436
AgsRecallChannelRun:source, 1436
AgsRecallChannelRunClass, 1435

AgsRecallClass, 1404
AgsRecallContainer, 1358
AgsRecallContainer:recall-audio, 1358
AgsRecallContainer:recall-audio-run, 1358
AgsRecallContainer:recall-audio-run-type, 1359
AgsRecallContainer:recall-audio-type, 1359
AgsRecallContainer:recall-channel, 1359
AgsRecallContainer:recall-channel-run, 1359
AgsRecallContainer:recall-channel-run-type, 1359
AgsRecallContainer:recall-channel-type, 1360
AgsRecallContainerClass, 1358
AgsRecallContainerFindFlags, 1357
AgsRecallContainerFlags, 1357
AgsRecallDependency, 1363
AgsRecallDependency:dependency, 1364
AgsRecallDependencyClass, 1363
AgsRecallFlags, 1402
AgsRecallHandler, 1403
AgsRecallID, 1342
AgsRecallID:recycling-context, 1342
AgsRecallIDClass, 1342
AgsRecallNotifyDependencyMode, 1402
AgsRecallRecycling, 1439
AgsRecallRecycling:audio-channel, 1439
AgsRecallRecycling:child-destination, 1440
AgsRecallRecycling:child-source, 1440
AgsRecallRecycling:destination, 1440
AgsRecallRecycling:source, 1440
AgsRecallRecyclingClass, 1439
AgsRecallRecyclingFlags, 1439
AGS_RECYCLING, 1117
AgsRecycling, 1117
AgsRecycling::add-audio-signal, 1120
AgsRecycling::data-request, 1121
AgsRecycling::remove-audio-signal, 1121
AgsRecycling:audio-signal, 1118
AgsRecycling:buffer-size, 1118
AgsRecycling:channel, 1118
AgsRecycling:format, 1119
AgsRecycling:input-soundcard, 1119
AgsRecycling:input-soundcard-channel, 1119
AgsRecycling:next, 1119
AgsRecycling:output-soundcard, 1119
AgsRecycling:output-soundcard-channel, 1120
AgsRecycling:parent, 1120
AgsRecycling:prev, 1120
AgsRecycling:samplerate, 1120
ags_recycling_add_audio_signal, 1113
AGS_RECYCLING_CLASS, 1117
AGS_RECYCLING_CONTEXT, 1349
ags_recycling_context_add, 1345
ags_recycling_context_add_child, 1347
AGS_RECYCLING_CONTEXT_CLASS, 1349
ags_recycling_context_find, 1346
ags_recycling_context_find_child, 1346
ags_recycling_context_find_parent, 1347
ags_recycling_context_find_scope, 1344
ags_recycling_context_flags_get_type, 1349
ags_recycling_context_get_child_recall_id, 1348
AGS_RECYCLING_CONTEXT_GET_CLASS, 1349
AGS_RECYCLING_CONTEXT_GET_OBJ_MUTEX, 1344
ags_recycling_context_get_toplevel, 1346
ags_recycling_context_get_type, 1350
ags_recycling_context_insert, 1345
ags_recycling_context_new, 1349
ags_recycling_context_remove, 1345
ags_recycling_context_remove_child, 1347
ags_recycling_context_replace, 1344
ags_recycling_context_reset_recycling, 1348
ags_recycling_create_audio_signal_with_defaults, 1115
ags_recycling_create_audio_signal_with_frame_count, 1116
ags_recycling_data_request, 1114
ags_recycling_find_next_channel, 1114
ags_recycling_flags_get_type, 1117
ags_recycling_get_audio_signal, 1112
ags_recycling_get_buffer_size, 1111
ags_recycling_get_channel, 1108
AGS_RECYCLING_GET_CLASS, 1117
ags_recycling_get_format, 1112
ags_recycling_get_input_soundcard, 1110
AGS_RECYCLING_GET_OBJ_MUTEX, 1106
ags_recycling_get_obj_mutex, 1107
ags_recycling_get_output_soundcard, 1109
ags_recycling_get_samplerate, 1111
ags_recycling_get_type, 1117
ags_recycling_is_active, 1115
ags_recycling_new, 1116
ags_recycling_next, 1109
ags_recycling_position, 1114
ags_recycling_prev, 1109
ags_recycling_remove_audio_signal, 1113
ags_recycling_set_audio_signal, 1113
ags_recycling_set_buffer_size, 1112
ags_recycling_set_channel, 1108
ags_recycling_set_flags, 1107
ags_recycling_set_format, 1112
ags_recycling_set_input_soundcard, 1110
ags_recycling_set_output_soundcard, 1110
ags_recycling_set_samplerate, 1111
ags_recycling_test_flags, 1107
ags_recycling_unset_flags, 1108
AgsRecyclingClass, 1118
AgsRecyclingContext, 1350
AgsRecyclingContext:child, 1350
AgsRecyclingContext:length, 1351
AgsRecyclingContext:parent, 1351
AgsRecyclingContext:recall-id, 1351
AgsRecyclingContextClass, 1350
AgsRecyclingContextFlags, 1350
AgsRecyclingFlags, 1117
AGS_REMOVE_AUDIO, 2368
AGS_REMOVE_AUDIO_CLASS, 2368
AGS_REMOVE_AUDIO_GET_CLASS, 2368
ags_remove_audio_get_type, 2368

ags_remove_audio_new, 2367
AGS_REMOVE_AUDIO_SIGNAL, 2370
AGS_REMOVE_AUDIO_SIGNAL_CLASS, 2370
AGS_REMOVE_AUDIO_SIGNAL_GET_CLASS, 2370
ags_remove_audio_signal_get_type, 2371
ags_remove_audio_signal_new, 2370
AGS_REMOVE_NOTE, 2373
AGS_REMOVE_NOTE_CLASS, 2373
AGS_REMOVE_NOTE_GET_CLASS, 2373
ags_remove_note_get_type, 2373
ags_remove_note_new, 2372
AGS_REMOVE_SOUNDCARD, 2376
AGS_REMOVE_SOUNDCARD_CLASS, 2376
AGS_REMOVE_SOUNDCARD_GET_CLASS, 2376
ags_remove_soundcard_get_type, 2377
ags_remove_soundcard_new, 2376
AgsRemoveAudio, 2368
AgsRemoveAudio:audio, 2368
AgsRemoveAudioClass, 2368
AgsRemoveAudioSignal, 2371
AgsRemoveAudioSignal:audio-signal, 2371
AgsRemoveAudioSignal:recycling, 2371
AgsRemoveAudioSignalClass, 2371
AgsRemoveNote, 2374
AgsRemoveNote:audio, 2374
AgsRemoveNote:audio-channel, 2374
AgsRemoveNote:note, 2374
AgsRemoveNote:use-selection-list, 2375
AgsRemoveNoteClass, 2374
AgsRemoveSoundcard, 2377
AgsRemoveSoundcard:soundcard, 2377
AgsRemoveSoundcardClass, 2377
AGS_RESAMPLE_UTIL, 374
ags_resample_util_alloc, 366
ags_resample_util_compute, 374
ags_resample_util_compute_complex, 374
ags_resample_util_compute_double, 373
ags_resample_util_compute_float, 373
ags_resample_util_compute_s16, 372
ags_resample_util_compute_s24, 372
ags_resample_util_compute_s32, 373
ags_resample_util_compute_s64, 373
ags_resample_util_compute_s8, 372
ags_resample_util_copy, 366
AGS_RESAMPLE_UTIL_DEFAULT_AUDIO_BUFFER_UTIL_FORMAT, 375
AGS_RESAMPLE_UTIL_DEFAULT_FORMAT, 374
AGS_RESAMPLE_UTIL_DEFAULT_SAMPLERATE, 375
AGS_RESAMPLE_UTIL_DEFAULT_TARGET_SAMPLERATE, 375
ags_resample_util_free, 367
ags_resample_util_get_buffer_length, 369
ags_resample_util_get_destination, 367
ags_resample_util_get_destination_stride, 367
ags_resample_util_get_format, 370
ags_resample_util_get_samplerate, 371
ags_resample_util_get_source, 368
ags_resample_util_get_source_stride, 369
ags_resample_util_get_target_samplerate, 371
ags_resample_util_get_type, 374
ags_resample_util_set_buffer_length, 370
ags_resample_util_set_destination, 367
ags_resample_util_set_destination_stride, 368
ags_resample_util_set_format, 370
ags_resample_util_set_samplerate, 371
ags_resample_util_set_source, 368
ags_resample_util_set_source_stride, 369
ags_resample_util_set_target_samplerate, 372
AgsResampleUtil, 375
AGS_RESET_FX_ANALYSE, 2386
ags_reset_fx_analyse_add, 2384
AGS_RESET_FX_ANALYSE_CLASS, 2386
AGS_RESET_FX_ANALYSE_GET_CLASS, 2386
ags_reset_fx_analyse_get_instance, 2385
ags_reset_fx_analyse_get_type, 2386
ags_reset_fx_analyse_new, 2385
ags_reset_fx_analyse_remove, 2385
AGS_RESET_FX_PEAK, 2383
ags_reset_fx_peak_add, 2381
AGS_RESET_FX_PEAK_CLASS, 2383
AGS_RESET_FX_PEAK_GET_CLASS, 2383
ags_reset_fx_peak_get_instance, 2382
ags_reset_fx_peak_get_type, 2383
ags_reset_fx_peak_new, 2382
ags_reset_fx_peak_remove, 2382
AGS_RESET_NOTE, 2380
ags_reset_note_add, 2378
AGS_RESET_NOTE_CLASS, 2380
AGS_RESET_NOTE_GET_CLASS, 2380
ags_reset_note_get_instance, 2379
ags_reset_note_get_type, 2380
ags_reset_note_new, 2379
ags_reset_note_remove, 2379
AgsResetFxAnalyse, 2386
AgsResetFxAnalyseClass, 2386
AgsResetFxPeak, 2383
AgsResetFxPeakClass, 2383
AgsResetNote, 2380
AgsResetNoteClass, 2380
AGS_RESIZE_AUDIO, 2388
AGS_RESIZE_AUDIO_CLASS, 2388
AGS_RESIZE_AUDIO_GET_CLASS, 2388
ags_resize_audio_get_type, 2388
ags_resize_audio_new, 2387
AgsResizeAudio, 2389
AgsResizeAudio:audio, 2389
AgsResizeAudio:audio-channels, 2389
AgsResizeAudio:input-pads, 2389
AgsResizeAudio:output-pads, 2390
AgsResizeAudioClass, 2389

S

AGS_SEEK_SOUNDCARD, 2391
AGS_SEEK_SOUNDCARD_CLASS, 2392

AGS_SEEK_SOUNDCARD_GET_CLASS, 2392
ags_seek_soundcard_get_type, 2392
ags_seek_soundcard_new, 2391
AgsSeekSoundcard, 2392
AgsSeekSoundcard:offset, 2392
AgsSeekSoundcard:soundcard, 2393
AgsSeekSoundcard:whence, 2393
AgsSeekSoundcardClass, 2392
AGS_SEQUENCER_THREAD, 1480
AGS_SEQUENCER_THREAD_CLASS, 1480
AGS_SEQUENCER_THREAD_DEFAULT_JIFFIE, 1480
ags_sequencer_thread_find_sequencer, 1479
AGS_SEQUENCER_THREAD_GET_CLASS, 1480
ags_sequencer_thread_get_type, 1480
ags_sequencer_thread_interval_timeout, 1479
ags_sequencer_thread_new, 1479
ags_sequencer_util_get_obj_mutex, 161
ags_sequencer_util_get_type, 161
AgsSequencerThread, 1480
AgsSequencerThread::interval-timeout, 1481
AgsSequencerThread:sequencer, 1481
AgsSequencerThreadClass, 1480
AgsSequencerUtil, 161
AGS_SET_AUDIO_CHANNELS, 2395
AGS_SET_AUDIO_CHANNELS_CLASS, 2395
AGS_SET_AUDIO_CHANNELS_GET_CLASS, 2395
ags_set_audio_channels_get_type, 2395
ags_set_audio_channels_new, 2394
AGS_SET_BUFFER_SIZE, 2398
AGS_SET_BUFFER_SIZE_CLASS, 2398
AGS_SET_BUFFER_SIZE_GET_CLASS, 2398
ags_set_buffer_size_get_type, 2398
ags_set_buffer_size_new, 2397
AGS_SET_DEVICE, 2400
AGS_SET_DEVICE_CLASS, 2400
AGS_SET_DEVICE_GET_CLASS, 2400
ags_set_device_get_type, 2401
ags_set_device_new, 2400
AGS_SET_FORMAT, 2403
AGS_SET_FORMAT_CLASS, 2403
AGS_SET_FORMAT_GET_CLASS, 2403
ags_set_format_get_type, 2403
ags_set_format_new, 2402
AGS_SET_MUTED, 2406
AGS_SET_MUTED_CLASS, 2406
AGS_SET_MUTED_GET_CLASS, 2406
ags_set_muted_get_type, 2406
ags_set_muted_new, 2405
AGS_SET_SAMPLERATE, 2409
AGS_SET_SAMPLERATE_CLASS, 2409
AGS_SET_SAMPLERATE_GET_CLASS, 2409
ags_set_samplerate_get_type, 2409
ags_set_samplerate_new, 2408
AgsSetAudioChannels, 2395
AgsSetAudioChannels:audio-channels, 2396
AgsSetAudioChannels:soundcard, 2396
AgsSetAudioChannelsClass, 2395
AgsSetBufferSize, 2398
AgsSetBufferSize:buffer-size, 2398
AgsSetBufferSize:scope, 2399
AgsSetBufferSizeClass, 2398
AgsSetDevice, 2401
AgsSetDevice:device, 2401
AgsSetDevice:scope, 2401
AgsSetDeviceClass, 2401
AgsSetFormat, 2404
AgsSetFormat:format, 2404
AgsSetFormat:scope, 2404
AgsSetFormatClass, 2404
AgsSetMuted, 2407
AgsSetMuted:muted, 2407
AgsSetMuted:scope, 2407
AgsSetMutedClass, 2407
AgsSetSamplerate, 2410
AgsSetSamplerate:samplerate, 2410
AgsSetSamplerate:scope, 2410
AgsSetSamplerateClass, 2410
AGS_SF2_LOADER, 1491
AGS_SF2_LOADER_CLASS, 1491
ags_sf2_loader_get_audio, 1487
ags_sf2_loader_get_audio_container, 1489
AGS_SF2_LOADER_GET_CLASS, 1491
ags_sf2_loader_get_filename, 1487
ags_sf2_loader_get_instrument, 1489
AGS_SF2_LOADER_GET_OBJ_MUTEX, 1486
ags_sf2_loader_get_preset, 1488
ags_sf2_loader_get_type, 1491
ags_sf2_loader_new, 1490
ags_sf2_loader_set_audio, 1487
ags_sf2_loader_set_audio_container, 1490
ags_sf2_loader_set_filename, 1488
ags_sf2_loader_set_flags, 1486
ags_sf2_loader_set_instrument, 1489
ags_sf2_loader_set_preset, 1488
ags_sf2_loader_start, 1490
ags_sf2_loader_test_flags, 1486
ags_sf2_loader_unset_flags, 1486
AGS_SF2_MIDI_LOCALE_LOADER, 1501
AGS_SF2_MIDI_LOCALE_LOADER_CLASS, 1502
ags_sf2_midi_locale_loader_get_audio, 1496
ags_sf2_midi_locale_loader_get_audio_container, 1498
ags_sf2_midi_locale_loader_get_bank, 1497
AGS_SF2_MIDI_LOCALE_LOADER_GET_CLASS, 1502
ags_sf2_midi_locale_loader_get_filename, 1496
AGS_SF2_MIDI_LOCALE_LOADER_GET_OBJ_MUTEX, 1495
ags_sf2_midi_locale_loader_get_program, 1498
ags_sf2_midi_locale_loader_get_synth, 1500
ags_sf2_midi_locale_loader_get_template, 1499
ags_sf2_midi_locale_loader_get_type, 1502
ags_sf2_midi_locale_loader_new, 1501
ags_sf2_midi_locale_loader_set_audio, 1496
ags_sf2_midi_locale_loader_set_audio_container, 1499
ags_sf2_midi_locale_loader_set_bank, 1497

ags_sf2_midi_locale_loader_set_filename, 1497
ags_sf2_midi_locale_loader_set_flags, 1495
ags_sf2_midi_locale_loader_set_program, 1498
ags_sf2_midi_locale_loader_set_synth, 1500
ags_sf2_midi_locale_loader_set_template, 1500
ags_sf2_midi_locale_loader_start, 1501
ags_sf2_midi_locale_loader_test_flags, 1495
ags_sf2_midi_locale_loader_unset_flags, 1495
AGS_SF2_SYNTH_GENERATOR, 529
AGS_SF2_SYNTH_GENERATOR_CLASS, 529
ags_sf2_synth_generator_compute, 528
ags_sf2_synth_generator_compute_instrument, 528
ags_sf2_synth_generator_compute_midi_locale, 528
AGS_SF2_SYNTH_GENERATOR_DEFAULT_BANK, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_BASE_KEY, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_FORMAT, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_PROGRAM, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_SAMPLERATE, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_TUNING, 530
AGS_SF2_SYNTH_GENERATOR_DEFAULT_VOLUME, 530
ags_sf2_synth_generator_get_bank, 519
ags_sf2_synth_generator_get_base_key, 526
ags_sf2_synth_generator_get_buffer_size, 521
AGS_SF2_SYNTH_GENERATOR_GET_CLASS, 529
ags_sf2_synth_generator_get_delay, 523
ags_sf2_synth_generator_get_filename, 517
ags_sf2_synth_generator_get_format, 522
ags_sf2_synth_generator_get_frame_count, 524
ags_sf2_synth_generator_get_instrument, 518
ags_sf2_synth_generator_get_loop_end, 525
ags_sf2_synth_generator_get_loop_start, 524
AGS_SF2_SYNTH_GENERATOR_GET_OBJ_MUTEX, 516
ags_sf2_synth_generator_get_obj_mutex, 516
ags_sf2_synth_generator_get_pitch_type, 522
ags_sf2_synth_generator_get_preset, 518
ags_sf2_synth_generator_get_program, 520
ags_sf2_synth_generator_get_samplerate, 520
ags_sf2_synth_generator_get_timestamp, 527
ags_sf2_synth_generator_get_tuning, 526
ags_sf2_synth_generator_get_type, 530
ags_sf2_synth_generator_new, 529
ags_sf2_synth_generator_set_bank, 519
ags_sf2_synth_generator_set_base_key, 526
ags_sf2_synth_generator_set_buffer_size, 521
ags_sf2_synth_generator_set_delay, 523
ags_sf2_synth_generator_set_filename, 517
ags_sf2_synth_generator_set_flags, 516
ags_sf2_synth_generator_set_format, 522
ags_sf2_synth_generator_set_frame_count, 524
ags_sf2_synth_generator_set_instrument, 519
ags_sf2_synth_generator_set_loop_end, 525
ags_sf2_synth_generator_set_loop_start, 525
ags_sf2_synth_generator_set_pitch_type, 523
ags_sf2_synth_generator_set_preset, 518
ags_sf2_synth_generator_set_program, 520
ags_sf2_synth_generator_set_samplerate, 521
ags_sf2_synth_generator_set_timestamp, 527
ags_sf2_synth_generator_set_tuning, 527
ags_sf2_synth_generator_test_flags, 516
ags_sf2_synth_generator_unset_flags, 517
AGS_SF2_SYNTH_UTIL, 448
ags_sf2_synth_util_alloc, 429
ags_sf2_synth_util_boxed_copy, 429
ags_sf2_synth_util_compute, 447
ags_sf2_synth_util_compute_complex, 446
ags_sf2_synth_util_compute_double, 446
ags_sf2_synth_util_compute_float, 446
ags_sf2_synth_util_compute_s16, 445
ags_sf2_synth_util_compute_s24, 445
ags_sf2_synth_util_compute_s32, 445
ags_sf2_synth_util_compute_s64, 446
ags_sf2_synth_util_compute_s8, 445
ags_sf2_synth_util_free, 430
ags_sf2_synth_util_get_bank, 436
ags_sf2_synth_util_get_buffer_length, 432
ags_sf2_synth_util_get_format, 432
ags_sf2_synth_util_get_frame_count, 439
ags_sf2_synth_util_get_instrument, 434
ags_sf2_synth_util_get_loop_end, 441
ags_sf2_synth_util_get_loop_mode, 440
ags_sf2_synth_util_get_loop_start, 441
ags_sf2_synth_util_get_midi_key, 437
ags_sf2_synth_util_get_note, 437
ags_sf2_synth_util_get_offset, 439
ags_sf2_synth_util_get_pitch_type, 443
ags_sf2_synth_util_get_pitch_util, 443
ags_sf2_synth_util_get_preset, 434
ags_sf2_synth_util_get_program, 436
ags_sf2_synth_util_get_resample_util, 442
ags_sf2_synth_util_get_sample, 435
ags_sf2_synth_util_get_samplerate, 433
ags_sf2_synth_util_get_sf2_file, 430
ags_sf2_synth_util_get_source, 431
ags_sf2_synth_util_get_source_stride, 431
ags_sf2_synth_util_get_type, 448
ags_sf2_synth_util_get_volume, 438
ags_sf2_synth_util_load_instrument, 444
ags_sf2_synth_util_load_midi_locale, 444
ags_sf2_synth_util_loop_mode_get_type, 448
ags_sf2_synth_util_midi_locale_find_sample_near_midi_key, 447
ags_sf2_synth_util_set_bank, 436
ags_sf2_synth_util_set_buffer_length, 432
ags_sf2_synth_util_set_format, 433
ags_sf2_synth_util_set_frame_count, 439

ags_sf2_synth_util_set_instrument, 435
ags_sf2_synth_util_set_loop_end, 442
ags_sf2_synth_util_set_loop_mode, 440
ags_sf2_synth_util_set_loop_start, 441
ags_sf2_synth_util_set_midi_key, 437
ags_sf2_synth_util_set_note, 438
ags_sf2_synth_util_set_offset, 440
ags_sf2_synth_util_set_pitch_type, 443
ags_sf2_synth_util_set_pitch_util, 444
ags_sf2_synth_util_set_preset, 434
ags_sf2_synth_util_set_program, 437
ags_sf2_synth_util_set_resample_util, 442
ags_sf2_synth_util_set_sample, 435
ags_sf2_synth_util_set_samplerate, 433
ags_sf2_synth_util_set_sf2_file, 430
ags_sf2_synth_util_set_source, 431
ags_sf2_synth_util_set_source_stride, 432
ags_sf2_synth_util_set_volume, 438
AgsSF2Level, 1845
AgsSF2Loader, 1492
AgsSF2Loader:audio, 1492
AgsSF2Loader:audio-container, 1492
AgsSF2Loader:filename, 1493
AgsSF2Loader:instrument, 1493
AgsSF2Loader:preset, 1493
AgsSF2LoaderClass, 1492
AgsSF2LoaderFlags, 1491
AgsSF2MidiLocaleLoader, 1502
AgsSF2MidiLocaleLoader:audio, 1503
AgsSF2MidiLocaleLoader:audio-container, 1503
AgsSF2MidiLocaleLoader:bank, 1503
AgsSF2MidiLocaleLoader:filename, 1503
AgsSF2MidiLocaleLoader:program, 1504
AgsSF2MidiLocaleLoader:synth, 1504
AgsSF2MidiLocaleLoader:template, 1504
AgsSF2MidiLocaleLoaderClass, 1502
AgsSF2MidiLocaleLoaderFlags, 1502
AgsSF2SynthGenerator, 531
AgsSF2SynthGenerator:attack, 531
AgsSF2SynthGenerator:bank, 531
AgsSF2SynthGenerator:base-key, 532
AgsSF2SynthGenerator:buffer-size, 532
AgsSF2SynthGenerator:delay, 532
AgsSF2SynthGenerator:filename, 532
AgsSF2SynthGenerator:format, 533
AgsSF2SynthGenerator:frame-count, 533
AgsSF2SynthGenerator:instrument, 533
AgsSF2SynthGenerator:loop-end, 533
AgsSF2SynthGenerator:loop-start, 533
AgsSF2SynthGenerator:pitch-type, 534
AgsSF2SynthGenerator:preset, 534
AgsSF2SynthGenerator:program, 534
AgsSF2SynthGenerator:samplerate, 534
AgsSF2SynthGenerator:tuning, 535
AgsSF2SynthGenerator:volume, 535
AgsSF2SynthGeneratorClass, 531
AgsSF2SynthGeneratorFlags, 531
AgsSF2SynthUtil, 449
AgsSF2SynthUtilFlags, 448
AgsSF2SynthUtilLoopMode, 448
AGS_SFZ_FILE, 1868
ags_sfz_file_check_suffix, 1867
AGS_SFZ_FILE_CLASS, 1868
AGS_SFZ_FILE_DEFAULT_CHANNELS, 1869
AGS_SFZ_FILE_GET_CLASS, 1868
ags_sfz_file_get_group, 1865
AGS_SFZ_FILE_GET_OBJ_MUTEX, 1863
ags_sfz_file_get_range, 1867
ags_sfz_file_get_region, 1865
ags_sfz_file_get_sample, 1866
ags_sfz_file_get_type, 1869
AGS_SFZ_FILE_LOOP_MAX, 1869
ags_sfz_file_new, 1868
ags_sfz_file_parse, 1868
AGS_SFZ_FILE_READ, 1869
ags_sfz_file_select_sample, 1867
ags_sfz_file_set_flags, 1864
ags_sfz_file_set_group, 1865
ags_sfz_file_set_region, 1866
ags_sfz_file_set_sample, 1866
ags_sfz_file_test_flags, 1864
ags_sfz_file_unset_flags, 1864
AGS_SFZ_FILE_WRITE, 1869
AGS_SFZ_GROUP, 1876
AGS_SFZ_GROUP_CLASS, 1876
AGS_SFZ_GROUP_GET_CLASS, 1876
ags_sfz_group_get_control, 1875
AGS_SFZ_GROUP_GET_OBJ_MUTEX, 1872
ags_sfz_group_get_region, 1873
ags_sfz_group_get_sample, 1874
ags_sfz_group_get_type, 1876
ags_sfz_group_insert_control, 1875
ags_sfz_group_lookup_control, 1875
ags_sfz_group_new, 1876
ags_sfz_group_set_flags, 1873
ags_sfz_group_set_region, 1874
ags_sfz_group_set_sample, 1874
ags_sfz_group_test_flags, 1872
ags_sfz_group_unset_flags, 1873
AGS_SFZ_INSTRUMENT_LOADER, 1518
AGS_SFZ_INSTRUMENT_LOADER_CLASS, 1518
ags_sfz_instrument_loader_get_audio, 1514
ags_sfz_instrument_loader_get_audio_container, 1515
AGS_SFZ_INSTRUMENT_LOADER_GET_CLASS, 1518
ags_sfz_instrument_loader_get_filename, 1514
AGS_SFZ_INSTRUMENT_LOADER_GET_OBJ_MUTEX, 1512
ags_sfz_instrument_loader_get_synth, 1516
ags_sfz_instrument_loader_get_template, 1516
ags_sfz_instrument_loader_get_type, 1518
ags_sfz_instrument_loader_new, 1517
ags_sfz_instrument_loader_set_audio, 1514
ags_sfz_instrument_loader_set_audio_container, 1515
ags_sfz_instrument_loader_set_filename, 1515

ags_sfz_instrument_loader_set_flags, 1513
ags_sfz_instrument_loader_set_synth, 1517
ags_sfz_instrument_loader_set_template, 1516
ags_sfz_instrument_loader_start, 1517
ags_sfz_instrument_loader_test_flags, 1513
ags_sfz_instrument_loader_unset_flags, 1513
AGS_SFZ_LOADER, 1509
AGS_SFZ_LOADER_CLASS, 1509
ags_sfz_loader_get_audio, 1506
ags_sfz_loader_get_audio_container, 1508
AGS_SFZ_LOADER_GET_CLASS, 1509
ags_sfz_loader_get_filename, 1507
AGS_SFZ_LOADER_GET_OBJ_MUTEX, 1505
ags_sfz_loader_get_type, 1510
ags_sfz_loader_new, 1509
ags_sfz_loader_set_audio, 1507
ags_sfz_loader_set_audio_container, 1508
ags_sfz_loader_set_filename, 1507
ags_sfz_loader_set_flags, 1506
ags_sfz_loader_start, 1508
ags_sfz_loader_test_flags, 1505
ags_sfz_loader_unset_flags, 1506
AGS_SFZ_REGION, 1883
AGS_SFZ_REGION_CLASS, 1883
AGS_SFZ_REGION_GET_CLASS, 1883
ags_sfz_region_get_control, 1881
ags_sfz_region_get_group, 1880
AGS_SFZ_REGION_GET_OBJ_MUTEX, 1879
ags_sfz_region_get_sample, 1880
ags_sfz_region_get_type, 1883
ags_sfz_region_insert_control, 1881
ags_sfz_region_lookup_control, 1882
ags_sfz_region_new, 1882
ags_sfz_region_set_flags, 1879
ags_sfz_region_set_group, 1880
ags_sfz_region_set_sample, 1881
ags_sfz_region_test_flags, 1879
ags_sfz_region_unset_flags, 1879
AGS_SFZ_SAMPLE, 1891
AGS_SFZ_SAMPLE_CLASS, 1891
AGS_SFZ_SAMPLE_GET_CLASS, 1891
ags_sfz_sample_get_group, 1887
ags_sfz_sample_get_hikey, 1888
ags_sfz_sample_get_key, 1888
ags_sfz_sample_get_lokey, 1889
ags_sfz_sample_get_loop_end, 1890
ags_sfz_sample_get_loop_mode, 1889
ags_sfz_sample_get_loop_start, 1890
AGS_SFZ_SAMPLE_GET_OBJ_MUTEX, 1885
ags_sfz_sample_get_pitch_keycenter, 1889
ags_sfz_sample_get_region, 1887
ags_sfz_sample_get_type, 1891
ags_sfz_sample_new, 1890
ags_sfz_sample_set_flags, 1886
ags_sfz_sample_set_group, 1887
ags_sfz_sample_set_region, 1888
ags_sfz_sample_test_flags, 1886
ags_sfz_sample_unset_flags, 1886
AGS_SFZ_SYNTH_GENERATOR, 546
AGS_SFZ_SYNTH_GENERATOR_CLASS, 546
ags_sfz_synth_generator_compute, 545
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BASE_KEY, 547
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE, 547
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_FORMAT, 547
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_PITCH_TYPE, 547
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_SAMPLERATE, 546
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_TUNING, 547
AGS_SFZ_SYNTH_GENERATOR_DEFAULT_VOLUME, 547
ags_sfz_synth_generator_get_base_key, 543
ags_sfz_synth_generator_get_buffer_size, 539
AGS_SFZ_SYNTH_GENERATOR_GET_CLASS, 546
ags_sfz_synth_generator_get_delay, 541
ags_sfz_synth_generator_get_filename, 537
ags_sfz_synth_generator_get_format, 539
ags_sfz_synth_generator_get_frame_count, 541
ags_sfz_synth_generator_get_loop_end, 543
ags_sfz_synth_generator_get_loop_start, 542
AGS_SFZ_SYNTH_GENERATOR_GET_OBJ_MUTEX, 537
ags_sfz_synth_generator_get_obj_mutex, 537
ags_sfz_synth_generator_get_pitch_type, 540
ags_sfz_synth_generator_get_samplerate, 538
ags_sfz_synth_generator_get_timestamp, 545
ags_sfz_synth_generator_get_tuning, 544
ags_sfz_synth_generator_get_type, 546
ags_sfz_synth_generator_new, 546
ags_sfz_synth_generator_set_base_key, 544
ags_sfz_synth_generator_set_buffer_size, 539
ags_sfz_synth_generator_set_delay, 541
ags_sfz_synth_generator_set_filename, 538
ags_sfz_synth_generator_set_format, 540
ags_sfz_synth_generator_set_frame_count, 542
ags_sfz_synth_generator_set_loop_end, 543
ags_sfz_synth_generator_set_loop_start, 542
ags_sfz_synth_generator_set_pitch_type, 540
ags_sfz_synth_generator_set_samplerate, 538
ags_sfz_synth_generator_set_timestamp, 545
ags_sfz_synth_generator_set_tuning, 544
AGS_SFZ_SYNTH_UTIL, 465
ags_sfz_synth_util_alloc, 451
ags_sfz_synth_util_boxed_copy, 451
ags_sfz_synth_util_compute, 464
ags_sfz_synth_util_compute_complex, 464
ags_sfz_synth_util_compute_double, 464
ags_sfz_synth_util_compute_float, 464
ags_sfz_synth_util_compute_s16, 463
ags_sfz_synth_util_compute_s24, 463
ags_sfz_synth_util_compute_s32, 463

ags_sfz_synth_util_compute_s64, 463
ags_sfz_synth_util_compute_s8, 462
ags_sfz_synth_util_free, 452
ags_sfz_synth_util_get_buffer_length, 454
ags_sfz_synth_util_get_format, 454
ags_sfz_synth_util_get_frame_count, 458
ags_sfz_synth_util_get_loop_end, 460
ags_sfz_synth_util_get_loop_mode, 459
ags_sfz_synth_util_get_loop_start, 459
ags_sfz_synth_util_get_midi_key, 456
ags_sfz_synth_util_get_note, 456
ags_sfz_synth_util_get_offset, 458
ags_sfz_synth_util_get_pitch_type, 461
ags_sfz_synth_util_get_pitch_util, 461
ags_sfz_synth_util_get_samplerate, 455
ags_sfz_synth_util_get_sfz_file, 452
ags_sfz_synth_util_get_source, 453
ags_sfz_synth_util_get_source_stride, 453
ags_sfz_synth_util_get_type, 465
ags_sfz_synth_util_get_volume, 457
ags_sfz_synth_util_load_instrument, 462
ags_sfz_synth_util_loop_mode_get_type, 465
ags_sfz_synth_util_set_buffer_length, 454
ags_sfz_synth_util_set_format, 455
ags_sfz_synth_util_set_frame_count, 458
ags_sfz_synth_util_set_loop_end, 460
ags_sfz_synth_util_set_loop_mode, 459
ags_sfz_synth_util_set_loop_start, 460
ags_sfz_synth_util_set_midi_key, 456
ags_sfz_synth_util_set_note, 457
ags_sfz_synth_util_set_offset, 459
ags_sfz_synth_util_set_pitch_type, 461
ags_sfz_synth_util_set_pitch_util, 462
ags_sfz_synth_util_set_samplerate, 455
ags_sfz_synth_util_set_sfz_file, 452
ags_sfz_synth_util_set_source, 453
ags_sfz_synth_util_set_source_stride, 454
ags_sfz_synth_util_set_volume, 457
AgsSFZFile, 1869
AgsSFZFile:filename, 1870
AgsSFZFile:group, 1870
AgsSFZFile:mode, 1870
AgsSFZFile:region, 1870
AgsSFZFile:sample, 1871
AgsSFZFile:soundcard, 1871
AgsSFZFileClass, 1869
AgsSFZGroup, 1877
AgsSFZGroup:nth-group, 1877
AgsSFZGroup:region, 1877
AgsSFZGroup:sample, 1877
AgsSFZGroupClass, 1877
AgsSFZInstrumentLoader, 1519
AgsSFZInstrumentLoader:audio, 1519
AgsSFZInstrumentLoader:audio-container, 1519
AgsSFZInstrumentLoader:filename, 1519
AgsSFZInstrumentLoader:synth, 1519
AgsSFZInstrumentLoader:template, 1520
AgsSFZInstrumentLoaderClass, 1519
AgsSFZInstrumentLoaderFlags, 1518
AgsSFZLevel, 1869
AgsSFZLoader, 1510
AgsSFZLoader:audio, 1510
AgsSFZLoader:audio-container, 1511
AgsSFZLoader:filename, 1511
AgsSFZLoaderClass, 1510
AgsSFZLoaderFlags, 1510
AgsSFZRegion, 1883
AgsSFZRegion:group, 1883
AgsSFZRegion:nth-region, 1884
AgsSFZRegion:sample, 1884
AgsSFZRegionClass, 1883
AgsSFZSample, 1892
AgsSFZSample:audio-channels, 1892
AgsSFZSample:buffer-size, 1893
AgsSFZSample:filename, 1893
AgsSFZSample:format, 1893
AgsSFZSample:group, 1893
AgsSFZSample:loop-end, 1894
AgsSFZSample:loop-start, 1894
AgsSFZSample:region, 1894
AgsSFZSampleClass, 1892
AgsSFZSampleFlags, 1891
AgsSFZSampleLoopMode, 1892
AgsSFZSynthGenerator, 547
AgsSFZSynthGenerator:attack, 548
AgsSFZSynthGenerator:base-key, 548
AgsSFZSynthGenerator:buffer-size, 548
AgsSFZSynthGenerator:delay, 548
AgsSFZSynthGenerator:filename, 549
AgsSFZSynthGenerator:format, 549
AgsSFZSynthGenerator:frame-count, 549
AgsSFZSynthGenerator:loop-end, 549
AgsSFZSynthGenerator:loop-start, 549
AgsSFZSynthGenerator:pitch-type, 550
AgsSFZSynthGenerator:samplerate, 550
AgsSFZSynthGenerator:tuning, 550
AgsSFZSynthGenerator:volume, 550
AgsSFZSynthGeneratorClass, 547
AgsSFZSynthUtil, 466
AgsSFZSynthUtilLoopMode, 465
AGS_SNDFILE, 1860
AgsSndfile, 1861
AgsSndfile:audio-channels, 1861
AgsSndfile:buffer-size, 1861
AgsSndfile:file, 1862
AgsSndfile:format, 1862
ags_sndfile_check_suffix, 1859
AGS_SNDFILE_CLASS, 1860
ags_sndfile_flags_get_type, 1860
AGS_SNDFILE_GET_CLASS, 1860
AGS_SNDFILE_GET_OBJ_MUTEX, 1858
ags_sndfile_get_type, 1860
ags_sndfile_new, 1859
ags_sndfile_set_flags, 1858

ags_sndfile_test_flags, 1858
ags_sndfile_unset_flags, 1859
AgsSndfileClass, 1861
AgsSndfileFlags, 1860
ags_sound_ability_flags_get_type, 152
ags_sound_behaviour_flags_get_type, 152
AGS_SOUND_CONTAINER, 1901
ags_sound_container_close, 1900
AGS_SOUND_CONTAINER_GET_INTERFACE, 1901
ags_sound_container_get_level_count, 1896
ags_sound_container_get_level_id, 1896
ags_sound_container_get_level_index, 1897
ags_sound_container_get_nesting_level, 1896
ags_sound_container_get_resource_all, 1899
ags_sound_container_get_resource_by_index, 1899
ags_sound_container_get_resource_by_name, 1899
ags_sound_container_get_resource_current, 1900
ags_sound_container_get_sublevel_name, 1897
ags_sound_container_get_type, 1901
AGS_SOUND_CONTAINER_INTERFACE, 1901
ags_sound_container_level_up, 1897
ags_sound_container_open, 1895
ags_sound_container_select_level_by_id, 1898
ags_sound_container_select_level_by_index, 1898
ags_sound_key_format_get_type, 152
AGS_SOUND_PROVIDER, 954
ags_sound_provider_get_audio, 952
ags_sound_provider_get_default_soundcard, 949
ags_sound_provider_get_default_soundcard_thread, 950
AGS_SOUND_PROVIDER_GET_INTERFACE, 954
ags_sound_provider_get_osc_server, 953
ags_sound_provider_get_sequencer, 951
ags_sound_provider_get_sound_server, 953
ags_sound_provider_get_soundcard, 951
ags_sound_provider_get_type, 954
AGS_SOUND_PROVIDER_INTERFACE, 954
ags_sound_provider_set_audio, 952
ags_sound_provider_set_default_soundcard, 949
ags_sound_provider_set_default_soundcard_thread, 950
ags_sound_provider_set_osc_server, 953
ags_sound_provider_set_sequencer, 951
ags_sound_provider_set_sound_server, 952
ags_sound_provider_set_soundcard, 950
AGS_SOUND_RESOURCE, 1909
ags_sound_resource_close, 1908
ags_sound_resource_flush, 1907
AGS_SOUND_RESOURCE_GET_INTERFACE, 1909
ags_sound_resource_get_presets, 1905
ags_sound_resource_get_type, 1909
ags_sound_resource_info, 1904
AGS_SOUND_RESOURCE_INTERFACE, 1909
ags_sound_resource_load, 1904
ags_sound_resource_open, 1903
ags_sound_resource_read, 1906
ags_sound_resource_read_audio_signal, 1907
ags_sound_resource_read_wave, 1908
ags_sound_resource_rw_open, 1904
ags_sound_resource_seek, 1907
ags_sound_resource_set_presets, 1905
ags_sound_resource_write, 1906
ags_sound_scope_get_type, 152
ags_sound_staging_flags_get_type, 152
ags_sound_state_flags_get_type, 153
AgsSoundAbilityFlags, 153
AgsSoundBehaviourFlags, 153
AGS_SOUNDCARD_THREAD, 1483
AGS_SOUNDCARD_THREAD_CLASS, 1483
AGS_SOUNDCARD_THREAD_DEFAULT_JIFFIE, 1484
ags_soundcard_thread_find_soundcard, 1482
AGS_SOUNDCARD_THREAD_GET_CLASS, 1483
ags_soundcard_thread_get_type, 1483
ags_soundcard_thread_new, 1483
ags_soundcard_util_adjust_delay_and_attack, 159
ags_soundcard_util_get_obj_mutex, 159
ags_soundcard_util_get_type, 160
AgsSoundcardThread, 1484
AgsSoundcardThread:soundcard, 1484
AgsSoundcardThread:soundcard-capability, 1484
AgsSoundcardThreadClass, 1484
AgsSoundcardUtil, 160
AgsSoundContainer, 1901
AgsSoundContainerInterface, 1901
AgsSoundKeyFormat, 156
AgsSoundProvider, 955
AgsSoundProviderInterface, 955
AgsSoundResource, 1909
AgsSoundResourceInterface, 1909
AgsSoundScope, 154
AgsSoundStagingFlags, 154
AgsSoundStateFlags, 155
AGS_START_AUDIO, 2412
AGS_START_AUDIO_CLASS, 2412
AGS_START_AUDIO_GET_CLASS, 2412
ags_start_audio_get_type, 2412
ags_start_audio_new, 2411
AGS_START_CHANNEL, 2415
AGS_START_CHANNEL_CLASS, 2415
AGS_START_CHANNEL_GET_CLASS, 2415
ags_start_channel_get_type, 2415
ags_start_channel_new, 2414
AGS_START_SEQUENCER, 2418
AGS_START_SEQUENCER_CLASS, 2418
AGS_START_SEQUENCER_GET_CLASS, 2418
ags_start_sequencer_get_type, 2418
ags_start_sequencer_new, 2417
AGS_START_SOUNDCARD, 2420
AGS_START_SOUNDCARD_CLASS, 2420
AGS_START_SOUNDCARD_GET_CLASS, 2420
ags_start_soundcard_get_type, 2420
ags_start_soundcard_new, 2419
AgsStartAudio, 2413
AgsStartAudio:audio, 2413
AgsStartAudio:sound-scope, 2413
AgsStartAudioClass, 2413

AgsStartChannel, 2416
AgsStartChannel:channel, 2416
AgsStartChannel:sound-scope, 2416
AgsStartChannelClass, 2416
AgsStartSequencer, 2418
AgsStartSequencerClass, 2418
AgsStartSoundcard, 2420
AgsStartSoundcardClass, 2420
AGS_STOP_SEQUENCER, 2422
AGS_STOP_SEQUENCER_CLASS, 2422
AGS_STOP_SEQUENCER_GET_CLASS, 2422
ags_stop_sequencer_get_type, 2422
ags_stop_sequencer_new, 2421
AGS_STOP_SOUNDCARD, 2424
AGS_STOP_SOUNDCARD_CLASS, 2424
AGS_STOP_SOUNDCARD_GET_CLASS, 2424
ags_stop_soundcard_get_type, 2424
ags_stop_soundcard_new, 2424
AGS_STOP_THREAD, 2426
AGS_STOP_THREAD_CLASS, 2426
AGS_STOP_THREAD_GET_CLASS, 2426
ags_stop_thread_get_type, 2427
ags_stop_thread_new, 2426
AgsStopSequencer, 2422
AgsStopSequencerClass, 2423
AgsStopSoundcard, 2425
AgsStopSoundcardClass, 2425
AgsStopThread, 2427
AgsStopThreadClass, 2427
ags_stream_alloc, 1126
ags_stream_free, 1127
ags_stream_slice_alloc, 1127
ags_stream_slice_free, 1127
AGS_SWITCH_BUFFER_FLAG, 2429
AGS_SWITCH_BUFFER_FLAG_CLASS, 2429
AGS_SWITCH_BUFFER_FLAG_GET_CLASS, 2429
ags_switch_buffer_flag_get_type, 2429
ags_switch_buffer_flag_new, 2428
AgsSwitchBufferFlag, 2429
AgsSwitchBufferFlag:device, 2429
AgsSwitchBufferFlagClass, 2429
AGS_SYNTH_GENERATOR, 506
AGS_SYNTH_GENERATOR_CLASS, 506
ags_synth_generator_compute, 506
AGS_SYNTH_GENERATOR_DEFAULT_BUFFER_SIZE, 507
AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_DEPTH, 508
AGS_SYNTH_GENERATOR_DEFAULT_FM_LFO_FREQUENCY, 508
AGS_SYNTH_GENERATOR_DEFAULT_FM_TUNING, 508
AGS_SYNTH_GENERATOR_DEFAULT_FORMAT, 507
AGS_SYNTH_GENERATOR_DEFAULT_FREQUENCY, 507
AGS_SYNTH_GENERATOR_DEFAULT_LFO_DEPTH, 507
AGS_SYNTH_GENERATOR_DEFAULT_OSCILLATOR, 507
AGS_SYNTH_GENERATOR_DEFAULT_PHASE, 507
AGS_SYNTH_GENERATOR_DEFAULT_SAMPLERATE, 507
AGS_SYNTH_GENERATOR_DEFAULT_TUNING, 508
AGS_SYNTH_GENERATOR_DEFAULT_VOLUME, 507
ags_synth_generator_get_buffer_size, 493
AGS_SYNTH_GENERATOR_GET_CLASS, 507
ags_synth_generator_get_delay, 495
ags_synth_generator_get_do_fm_synth, 502
ags_synth_generator_get_do_lfo, 497
ags_synth_generator_get_fm_lfo_depth, 504
ags_synth_generator_get_fm_lfo_frequency, 503
ags_synth_generator_get_fm_lfo_oscillator, 502
ags_synth_generator_get_fm_tuning, 504
ags_synth_generator_get_format, 494
ags_synth_generator_get_frame_count, 495
ags_synth_generator_get_frequency, 498
ags_synth_generator_get_lfo_depth, 500
ags_synth_generator_get_loop_end, 497
ags_synth_generator_get_loop_start, 496
AGS_SYNTH_GENERATOR_GET_OBJ_MUTEX, 492
ags_synth_generator_get_obj_mutex, 492
ags_synth_generator_get_oscillator, 498
ags_synth_generator_get_phase, 499
ags_synth_generator_get_samplerate, 493
ags_synth_generator_get_timestamp, 505
ags_synth_generator_get_tuning, 501
ags_synth_generator_get_type, 507
ags_synth_generator_get_volume, 500
ags_synth_generator_new, 506
ags_synth_generator_set_buffer_size, 494
ags_synth_generator_set_delay, 495
ags_synth_generator_set_do_fm_synth, 502
ags_synth_generator_set_do_lfo, 498
ags_synth_generator_set_fm_lfo_depth, 504
ags_synth_generator_set_fm_lfo_frequency, 503
ags_synth_generator_set_fm_lfo_oscillator, 503
ags_synth_generator_set_fm_tuning, 505
ags_synth_generator_set_format, 494
ags_synth_generator_set_frame_count, 496
ags_synth_generator_set_frequency, 499
ags_synth_generator_set_lfo_depth, 501
ags_synth_generator_set_loop_end, 497
ags_synth_generator_set_loop_start, 496
ags_synth_generator_set_oscillator, 498
ags_synth_generator_set_phase, 499
ags_synth_generator_set_samplerate, 493
ags_synth_generator_set_timestamp, 505
ags_synth_generator_set_tuning, 501
ags_synth_generator_set_volume, 500
ags_synth_oscillator_mode_get_type, 157
ags_synth_util_alloc, 377
ags_synth_util_compute_impulse, 400
ags_synth_util_compute_impulse_complex, 400
ags_synth_util_compute_impulse_double, 400
ags_synth_util_compute_impulse_float, 399
ags_synth_util_compute_impulse_s16, 398
ags_synth_util_compute_impulse_s24, 399

ags_synth_util_compute_impulse_s32, 399
ags_synth_util_compute_impulse_s64, 399
ags_synth_util_compute_impulse_s8, 398
ags_synth_util_compute_sawtooth, 393
ags_synth_util_compute_sawtooth_complex, 393
ags_synth_util_compute_sawtooth_double, 393
ags_synth_util_compute_sawtooth_float, 392
ags_synth_util_compute_sawtooth_s16, 391
ags_synth_util_compute_sawtooth_s24, 392
ags_synth_util_compute_sawtooth_s32, 392
ags_synth_util_compute_sawtooth_s64, 392
ags_synth_util_compute_sawtooth_s8, 391
ags_synth_util_compute_sin, 391
ags_synth_util_compute_sin_complex, 391
ags_synth_util_compute_sin_double, 390
ags_synth_util_compute_sin_float, 390
ags_synth_util_compute_sin_s16, 389
ags_synth_util_compute_sin_s24, 389
ags_synth_util_compute_sin_s32, 390
ags_synth_util_compute_sin_s64, 390
ags_synth_util_compute_sin_s8, 389
ags_synth_util_compute_square, 398
ags_synth_util_compute_square_complex, 398
ags_synth_util_compute_square_double, 397
ags_synth_util_compute_square_float, 397
ags_synth_util_compute_square_s16, 396
ags_synth_util_compute_square_s24, 396
ags_synth_util_compute_square_s32, 397
ags_synth_util_compute_square_s64, 397
ags_synth_util_compute_square_s8, 396
ags_synth_util_compute_triangle, 396
ags_synth_util_compute_triangle_complex, 395
ags_synth_util_compute_triangle_double, 395
ags_synth_util_compute_triangle_float, 395
ags_synth_util_compute_triangle_s16, 394
ags_synth_util_compute_triangle_s24, 394
ags_synth_util_compute_triangle_s32, 394
ags_synth_util_compute_triangle_s64, 395
ags_synth_util_compute_triangle_s8, 394
ags_synth_util_copy, 378
AGS_SYNTH_UTIL_DEFAULT_FREQUENCY, 401
ags_synth_util_free, 378
ags_synth_util_get_buffer_length, 379
ags_synth_util_get_format, 380
ags_synth_util_get_frame_count, 384
ags_synth_util_get_frequency, 382
ags_synth_util_get_offset, 385
ags_synth_util_get_phase, 383
ags_synth_util_get_samplerate, 381
ags_synth_util_get_source, 378
ags_synth_util_get_source_stride, 379
ags_synth_util_get_synth_oscillator_mode, 381
ags_synth_util_get_type, 400
ags_synth_util_get_volume, 383
ags_synth_util_get_xcross_count, 388
ags_synth_util_get_xcross_count_complex, 388
ags_synth_util_get_xcross_count_double, 388
ags_synth_util_get_xcross_count_float, 387
ags_synth_util_get_xcross_count_s16, 386
ags_synth_util_get_xcross_count_s24, 386
ags_synth_util_get_xcross_count_s32, 386
ags_synth_util_get_xcross_count_s64, 387
ags_synth_util_get_xcross_count_s8, 385
ags_synth_util_set_buffer_length, 380
ags_synth_util_set_format, 380
ags_synth_util_set_frame_count, 384
ags_synth_util_set_frequency, 382
ags_synth_util_set_offset, 385
ags_synth_util_set_phase, 383
ags_synth_util_set_samplerate, 381
ags_synth_util_set_source, 379
ags_synth_util_set_source_stride, 379
ags_synth_util_set_synth_oscillator_mode, 382
ags_synth_util_set_volume, 384
AgsSynthGenerator, 508
AgsSynthGenerator:attack, 508
AgsSynthGenerator:buffer-size, 509
AgsSynthGenerator:delay, 509
AgsSynthGenerator:do-fm-synth, 509
AgsSynthGenerator:do-lfo, 509
AgsSynthGenerator:fm-lfo-depth, 510
AgsSynthGenerator:fm-lfo-frequency, 510
AgsSynthGenerator:fm-lfo-oscillator, 510
AgsSynthGenerator:fm-tuning, 510
AgsSynthGenerator:format, 511
AgsSynthGenerator:frame-count, 511
AgsSynthGenerator:frequency, 511
AgsSynthGenerator:lfo-depth, 511
AgsSynthGenerator:loop-end, 512
AgsSynthGenerator:loop-start, 512
AgsSynthGenerator:oscillator, 512
AgsSynthGenerator:phase, 512
AgsSynthGenerator:samplerate, 513
AgsSynthGenerator:timestamp, 513
AgsSynthGenerator:tuning, 513
AgsSynthGenerator:volume, 513
AgsSynthGeneratorClass, 508
AgsSynthOscillatorMode, 157
AgsSynthUtil, 401

T

AGS_TIC_DEVICE, 2431
AGS_TIC_DEVICE_CLASS, 2431
AGS_TIC_DEVICE_GET_CLASS, 2431
ags_tic_device_get_type, 2431
ags_tic_device_new, 2430
AgsTicDevice, 2432
AgsTicDevice:device, 2432
AgsTicDeviceClass, 2432
AGS_TIME_STRETCH_UTIL, 245
ags_time_stretch_util_alloc, 234
ags_time_stretch_util_copy, 234
AGS_TIME_STRETCH_UTIL_DEFAULT_FREQUENCY, 245

ags_time_stretch_util_free, 234
ags_time_stretch_util_get_buffer_size, 239
ags_time_stretch_util_get_destination, 237
ags_time_stretch_util_get_destination_buffer_length, 238
ags_time_stretch_util_get_destination_stride, 237
ags_time_stretch_util_get_format, 239
ags_time_stretch_util_get_frequency, 240
ags_time_stretch_util_get_new_bpm, 242
ags_time_stretch_util_get_orig_bpm, 241
ags_time_stretch_util_get_samplerate, 240
ags_time_stretch_util_get_source, 235
ags_time_stretch_util_get_source_buffer_length, 236
ags_time_stretch_util_get_source_stride, 235
ags_time_stretch_util_get_type, 245
ags_time_stretch_util_set_buffer_size, 239
ags_time_stretch_util_set_destination, 237
ags_time_stretch_util_set_destination_buffer_length, 238
ags_time_stretch_util_set_destination_stride, 238
ags_time_stretch_util_set_format, 240
ags_time_stretch_util_set_frequency, 241
ags_time_stretch_util_set_new_bpm, 242
ags_time_stretch_util_set_orig_bpm, 241
ags_time_stretch_util_set_samplerate, 240
ags_time_stretch_util_set_source, 235
ags_time_stretch_util_set_source_buffer_length, 236
ags_time_stretch_util_set_source_stride, 236
ags_time_stretch_util_stretch, 244
ags_time_stretch_util_stretch_complex, 244
ags_time_stretch_util_stretch_double, 244
ags_time_stretch_util_stretch_float, 244
ags_time_stretch_util_stretch_s16, 243
ags_time_stretch_util_stretch_s24, 243
ags_time_stretch_util_stretch_s32, 243
ags_time_stretch_util_stretch_s64, 243
ags_time_stretch_util_stretch_s8, 242
AgsTimeStretchUtil, 245
AGS_TOGGLE_PATTERN_BIT, 2434
AGS_TOGGLE_PATTERN_BIT_CLASS, 2434
AGS_TOGGLE_PATTERN_BIT_GET_CLASS, 2434
ags_toggle_pattern_bit_get_type, 2434
ags_toggle_pattern_bit_new, 2433
ags_toggle_pattern_bit_refresh_gui, 2433
AgsTogglePatternBit, 2434
AgsTogglePatternBit:bit, 2435
AgsTogglePatternBit:index-i, 2435
AgsTogglePatternBit:index-j, 2435
AgsTogglePatternBit:line, 2435
AgsTogglePatternBit:pattern, 2436
AgsTogglePatternBitClass, 2435
AGS_TRACK, 1290
AgsTrack, 1291
AgsTrack:smf-buffer, 1291
AgsTrack:x, 1292
ags_track_alloc_smf_buffer, 1289
AGS_TRACK_CLASS, 1290
ags_track_duplicate, 1289
ags_track_flags_get_type, 1290
AGS_TRACK_GET_CLASS, 1290
AGS_TRACK_GET_OBJ_MUTEX, 1285
ags_track_get_obj_mutex, 1286
ags_track_get_smf_buffer, 1288
ags_track_get_type, 1291
ags_track_get_x, 1288
ags_track_lock, 1286
ags_track_new, 1290
ags_track_realloc_smf_buffer, 1289
ags_track_set_flags, 1287
ags_track_set_x, 1288
ags_track_sort_func, 1287
ags_track_test_flags, 1286
ags_track_unlock, 1286
ags_track_unset_flags, 1287
AgsTrackClass, 1291
AgsTrackFlags, 1291
AGS_TYPE_ACCELERATION, 1243
AGS_TYPE_ADD_AUDIO, 2272
AGS_TYPE_ADD_AUDIO_SIGNAL, 2275
AGS_TYPE_ADD_NOTE, 2278
AGS_TYPE_ADD_SOUNDCARD, 2281
AGS_TYPE_ALSA_DEVIN, 652
AGS_TYPE_ALSA_DEVOUT, 661
AGS_TYPE_ALSA_MIDIIN, 671
AGS_TYPE_APPLY_BPM, 2284
AGS_TYPE_APPLY_PRESETS, 2287
AGS_TYPE_APPLY_SEQUENCER_LENGTH, 2290
AGS_TYPE_APPLY_SF2_MIDI_LOCALE, 2303
AGS_TYPE_APPLY_SF2_SYNTH, 2300
AGS_TYPE_APPLY_SFZ_INSTRUMENT, 2310
AGS_TYPE_APPLY_SFZ_SYNTH, 2306
AGS_TYPE_APPLY_SOUND_CONFIG, 2293
AGS_TYPE_APPLY_SYNTH, 2296
AGS_TYPE_APPLY_TACT, 2312
AGS_TYPE_AUDIO, 1022
AGS_TYPE_AUDIO_APPLICATION_CONTEXT, 958
AGS_TYPE_AUDIO_BUFFER_UTIL, 322
AGS_TYPE_AUDIO_CONTAINER, 1790
AGS_TYPE_AUDIO_CONTAINER_MANAGER, 1780
AGS_TYPE_AUDIO_FILE, 1812
AGS_TYPE_AUDIO_FILE_LINK, 1817
AGS_TYPE_AUDIO_FILE_MANAGER, 1796
AGS_TYPE_AUDIO_LOOP, 1455
AGS_TYPE_AUDIO_SIGNAL, 1143
AGS_TYPE_AUDIO_THREAD, 1464
AGS_TYPE_AUDIO_UNIT_CLIENT, 919
AGS_TYPE_AUDIO_UNIT_DEVIN, 934
AGS_TYPE_AUDIO_UNIT_DEVOUT, 944
AGS_TYPE_AUDIO_UNIT_PORT, 926
AGS_TYPE_AUDIO_UNIT_SERVER, 911
AGS_TYPE_AUTOMATION, 1233
AGS_TYPE_BASE_PLUGIN, 18
AGS_TYPE_BUFFER, 1272
AGS_TYPE_CANCEL_AUDIO, 2315
AGS_TYPE_CANCEL_CHANNEL, 2318
AGS_TYPE_CHANNEL, 1083

AGS_TYPE_CHANNEL_THREAD, 1472
AGS_TYPE_CHAR_BUFFER_UTIL, 172
AGS_TYPE_CHORUS_UTIL, 189
AGS_TYPE_CLEAR_AUDIO_SIGNAL, 2321
AGS_TYPE_CLEAR_BUFFER, 2324
AGS_TYPE_CORE_AUDIO_CLIENT, 869
AGS_TYPE_CORE_AUDIO_DEVIN, 885
AGS_TYPE_CORE_AUDIO_DEVOUT, 894
AGS_TYPE_CORE_AUDIO_MIDIIN, 903
AGS_TYPE_CORE_AUDIO_PORT, 877
AGS_TYPE_CORE_AUDIO_SERVER, 862
AGS_TYPE_CROP_NOTE, 2327
AGS_TYPE_DSSI_MANAGER, 62
AGS_TYPE_DSSI_PLUGIN, 55
AGS_TYPE_ENVELOPE_UTIL, 341
AGS_TYPE_EXPORT_OUTPUT, 2332
AGS_TYPE_EXPORT_THREAD, 1477
AGS_TYPE_FAST_PITCH_UTIL, 221
AGS_TYPE_FIFOOUT, 642
AGS_TYPE_FLUID_IIR_FILTER_UTIL, 619
AGS_TYPE_FLUID_INTERPOLATE_4TH_ORDER_UTIL, 593
AGS_TYPE_FLUID_INTERPOLATE_7TH_ORDER_UTIL, 606
AGS_TYPE_FLUID_INTERPOLATE_LINEAR_UTIL, 579
AGS_TYPE_FLUID_INTERPOLATE_NONE_UTIL, 566
AGS_TYPE_FLUID_UTIL, 554
AGS_TYPE_FM_SYNTH_UTIL, 427
AGS_TYPE_FOURIER_TRANSFORM_UTIL, 210
AGS_TYPE_FREE_SELECTION, 2336
AGS_TYPE_FREQUENCY_ALIASE_UTIL, 266
AGS_TYPE_FREQUENCY_MAP, 632
AGS_TYPE_FREQUENCY_MAP_MANAGER, 627
AGS_TYPE_FX_ANALYSE_AUDIO, 2124
AGS_TYPE_FX_ANALYSE_AUDIO_PROCESSOR, 2126
AGS_TYPE_FX_ANALYSE_AUDIO_SIGNAL, 2129
AGS_TYPE_FX_ANALYSE_CHANNEL, 2134
AGS_TYPE_FX_ANALYSE_CHANNEL_PROCESSOR, 2137
AGS_TYPE_FX_ANALYSE_RECYCLING, 2139
AGS_TYPE_FX_BUFFER_AUDIO, 1937
AGS_TYPE_FX_BUFFER_AUDIO_PROCESSOR, 1939
AGS_TYPE_FX_BUFFER_AUDIO_SIGNAL, 1941
AGS_TYPE_FX_BUFFER_CHANNEL, 1946
AGS_TYPE_FX_BUFFER_CHANNEL_PROCESSOR, 1949
AGS_TYPE_FX_BUFFER_RECYCLING, 1951
AGS_TYPE_FX_DSSI_AUDIO, 1999
AGS_TYPE_FX_DSSI_AUDIO_PROCESSOR, 2001
AGS_TYPE_FX_DSSI_AUDIO_SIGNAL, 2004
AGS_TYPE_FX_DSSI_CHANNEL, 2006
AGS_TYPE_FX_DSSI_CHANNEL_PROCESSOR, 2009
AGS_TYPE_FX_DSSI_RECYCLING, 2011
AGS_TYPE_FX_ENVELOPE_AUDIO, 2204
AGS_TYPE_FX_ENVELOPE_AUDIO_PROCESSOR, 2206
AGS_TYPE_FX_ENVELOPE_AUDIO_SIGNAL, 2208
AGS_TYPE_FX_ENVELOPE_CHANNEL, 2211
AGS_TYPE_FX_ENVELOPE_CHANNEL_PROCESSOR, 2214
AGS_TYPE_FX_ENVELOPE_RECYCLING, 2217
AGS_TYPE_FX_EQ10_AUDIO, 2186
AGS_TYPE_FX_EQ10_AUDIO_PROCESSOR, 2188
AGS_TYPE_FX_EQ10_AUDIO_SIGNAL, 2191
AGS_TYPE_FX_EQ10_CHANNEL, 2194
AGS_TYPE_FX_EQ10_CHANNEL_PROCESSOR, 2199
AGS_TYPE_FX_EQ10_RECYCLING, 2201
AGS_TYPE_FX_FACTORY, 1321
AGS_TYPE_FX_FACTORY_CREATE_FLAGS, 1321
AGS_TYPE_FX_FM_SYNTH_AUDIO, 2065
AGS_TYPE_FX_FM_SYNTH_AUDIO_PROCESSOR, 2077
AGS_TYPE_FX_FM_SYNTH_AUDIO_SIGNAL, 2079
AGS_TYPE_FX_FM_SYNTH_CHANNEL, 2082
AGS_TYPE_FX_FM_SYNTH_CHANNEL_PROCESSOR, 2084
AGS_TYPE_FX_FM_SYNTH_RECYCLING, 2086
AGS_TYPE_FX_HIGH_PASS_AUDIO, 2171
AGS_TYPE_FX_HIGH_PASS_AUDIO_PROCESSOR, 2174
AGS_TYPE_FX_HIGH_PASS_AUDIO_SIGNAL, 2176
AGS_TYPE_FX_HIGH_PASS_CHANNEL, 2178
AGS_TYPE_FX_HIGH_PASS_CHANNEL_PROCESSOR, 2181
AGS_TYPE_FX_HIGH_PASS_RECYCLING, 2184
AGS_TYPE_FX_LADSPA_AUDIO, 2220
AGS_TYPE_FX_LADSPA_AUDIO_PROCESSOR, 2222
AGS_TYPE_FX_LADSPA_AUDIO_SIGNAL, 2225
AGS_TYPE_FX_LADSPA_CHANNEL, 2228
AGS_TYPE_FX_LADSPA_CHANNEL_PROCESSOR, 2231
AGS_TYPE_FX_LADSPA_RECYCLING, 2233
AGS_TYPE_FX_LFO_AUDIO, 2235
AGS_TYPE_FX_LFO_AUDIO_PROCESSOR, 2238
AGS_TYPE_FX_LFO_AUDIO_SIGNAL, 2240
AGS_TYPE_FX_LFO_CHANNEL, 2243
AGS_TYPE_FX_LFO_CHANNEL_PROCESSOR, 2247
AGS_TYPE_FX_LFO_RECYCLING, 2249
AGS_TYPE_FX_LOW_PASS_AUDIO, 2157
AGS_TYPE_FX_LOW_PASS_AUDIO_PROCESSOR, 2159
AGS_TYPE_FX_LOW_PASS_AUDIO_SIGNAL, 2161
AGS_TYPE_FX_LOW_PASS_CHANNEL, 2164
AGS_TYPE_FX_LOW_PASS_CHANNEL_PROCESSOR, 2167
AGS_TYPE_FX_LOW_PASS_RECYCLING, 2169
AGS_TYPE_FX_LV2_AUDIO, 2019
AGS_TYPE_FX_LV2_AUDIO_PROCESSOR, 2021
AGS_TYPE_FX_LV2_AUDIO_SIGNAL, 2023
AGS_TYPE_FX_LV2_CHANNEL, 2027
AGS_TYPE_FX_LV2_CHANNEL_PROCESSOR, 2029
AGS_TYPE_FX_LV2_RECYCLING, 2032
AGS_TYPE_FX_NOTATION_AUDIO, 1975
AGS_TYPE_FX_NOTATION_AUDIO_PROCESSOR, 1981
AGS_TYPE_FX_NOTATION_AUDIO_SIGNAL, 1984
AGS_TYPE_FX_NOTATION_CHANNEL, 1987
AGS_TYPE_FX_NOTATION_CHANNEL_PROCESSOR, 1989
AGS_TYPE_FX_NOTATION_RECYCLING, 1991
AGS_TYPE_FX_PATTERN_AUDIO, 1954
AGS_TYPE_FX_PATTERN_AUDIO_PROCESSOR, 1959

AGS_TYPE_FX_PATTERN_AUDIO_SIGNAL, 1962
AGS_TYPE_FX_PATTERN_CHANNEL, 1965
AGS_TYPE_FX_PATTERN_CHANNEL_PROCESSOR, 1967
AGS_TYPE_FX_PATTERN_RECYCLING, 1969
AGS_TYPE_FX_PEAK_AUDIO, 2109
AGS_TYPE_FX_PEAK_AUDIO_PROCESSOR, 2111
AGS_TYPE_FX_PEAK_AUDIO_SIGNAL, 2114
AGS_TYPE_FX_PEAK_CHANNEL, 2117
AGS_TYPE_FX_PEAK_CHANNEL_PROCESSOR, 2119
AGS_TYPE_FX_PEAK_RECYCLING, 2122
AGS_TYPE_FX_PLAYBACK_AUDIO, 1918
AGS_TYPE_FX_PLAYBACK_AUDIO_PROCESSOR, 1925
AGS_TYPE_FX_PLAYBACK_AUDIO_SIGNAL, 1927
AGS_TYPE_FX_PLAYBACK_CHANNEL, 1930
AGS_TYPE_FX_PLAYBACK_CHANNEL_PROCESSOR, 1932
AGS_TYPE_FX_PLAYBACK_RECYCLING, 1934
AGS_TYPE_FX_SF2_SYNTH_AUDIO, 2092
AGS_TYPE_FX_SF2_SYNTH_AUDIO_PROCESSOR, 2097
AGS_TYPE_FX_SF2_SYNTH_AUDIO_SIGNAL, 2099
AGS_TYPE_FX_SF2_SYNTH_CHANNEL, 2101
AGS_TYPE_FX_SF2_SYNTH_CHANNEL_PROCESSOR, 2104
AGS_TYPE_FX_SF2_SYNTH_RECYCLING, 2106
AGS_TYPE_FX_SYNTH_AUDIO, 2038
AGS_TYPE_FX_SYNTH_AUDIO_PROCESSOR, 2050
AGS_TYPE_FX_SYNTH_AUDIO_SIGNAL, 2052
AGS_TYPE_FX_SYNTH_CHANNEL, 2055
AGS_TYPE_FX_SYNTH_CHANNEL_PROCESSOR, 2057
AGS_TYPE_FX_SYNTH_RECYCLING, 2059
AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO, 2252
AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_PROCESSOR, 2254
AGS_TYPE_FX_TWO_PASS_ALIASE_AUDIO_SIGNAL, 2256
AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL, 2263
AGS_TYPE_FX_TWO_PASS_ALIASE_CHANNEL_PROCESSOR, 2266
AGS_TYPE_FX_TWO_PASS_ALIASE_RECYCLING, 2269
AGS_TYPE_FX_VOLUME_AUDIO, 2142
AGS_TYPE_FX_VOLUME_AUDIO_PROCESSOR, 2144
AGS_TYPE_FX_VOLUME_AUDIO_SIGNAL, 2147
AGS_TYPE_FX_VOLUME_CHANNEL, 2149
AGS_TYPE_FX_VOLUME_CHANNEL_PROCESSOR, 2152
AGS_TYPE_FX_VOLUME_RECYCLING, 2154
AGS_TYPE_GENERIC_RECALL_CHANNEL_RUN, 1445
AGS_TYPE_GENERIC_RECALL_RECYCLING, 1448
AGS_TYPE_GSTREAMER_CLIENT, 827
AGS_TYPE_GSTREAMER_DEVIN, 842
AGS_TYPE_GSTREAMER_DEVOUT, 853
AGS_TYPE_GSTREAMER_FILE, 1825
AGS_TYPE_GSTREAMER_PORT, 834
AGS_TYPE_GSTREAMER_SERVER, 819
AGS_TYPE_HQ_PITCH_UTIL, 232
AGS_TYPE_INPUT, 1104
AGS_TYPE_IPATCH, 1831
AGS_TYPE_IPATCH_DLS2_READER, 1852
AGS_TYPE_IPATCH_GIG_READER, 1838
AGS_TYPE_IPATCH_SAMPLE, 1856
AGS_TYPE_IPATCH_SF2_READER, 1846
AGS_TYPE_JACK_CLIENT, 734
AGS_TYPE_JACK_DEVIN, 748
AGS_TYPE_JACK_DEVOUT, 758
AGS_TYPE_JACK_MIDIIN, 767
AGS_TYPE_JACK_PORT, 741
AGS_TYPE_JACK_SERVER, 726
AGS_TYPE_LADSPA_CONVERSION, 43
AGS_TYPE_LADSPA_MANAGER, 49
AGS_TYPE_LADSPA_PLUGIN, 52
AGS_TYPE_LFO_SYNTH_UTIL, 490
AGS_TYPE_LINEAR_INTERPOLATE_UTIL, 200
AGS_TYPE_LINK_CHANNEL, 2338
AGS_TYPE_LV2_CONVERSION, 66
AGS_TYPE_LV2_EVENT_MANAGER, 70
AGS_TYPE_LV2_LOG_MANAGER, 73
AGS_TYPE_LV2_MANAGER, 80
AGS_TYPE_LV2_OPTION_MANAGER, 87
AGS_TYPE_LV2_PLUGIN, 108
AGS_TYPE_LV2_PRESET, 118
AGS_TYPE_LV2_PRESET_MANAGER, 113
AGS_TYPE_LV2_TURTLE_PARSER, 91
AGS_TYPE_LV2_TURTLE_SCANNER, 97
AGS_TYPE_LV2_URI_MAP_MANAGER, 123
AGS_TYPE_LV2_URID_MANAGER, 127
AGS_TYPE_LV2_WORKER, 133
AGS_TYPE_LV2_WORKER_MANAGER, 137
AGS_TYPE_LV2UI_MANAGER, 143
AGS_TYPE_LV2UI_PLUGIN, 147
AGS_TYPE_MIDI, 1283
AGS_TYPE_MIDI_BUFFER_UTIL, 1552
AGS_TYPE_MIDI_BUILDER, 1595
AGS_TYPE_MIDI_FILE, 1565
AGS_TYPE_MIDI_PARSER, 1620
AGS_TYPE_MIDI_UTIL, 1577
AGS_TYPE_MOVE_NOTE, 2342
AGS_TYPE_NOISE_UTIL, 364
AGS_TYPE_NOTATION, 1183
AGS_TYPE_NOTE, 1204
AGS_TYPE_OPEN_FILE, 2347
AGS_TYPE_OPEN_SF2_INSTRUMENT, 2354
AGS_TYPE_OPEN_SF2_SAMPLE, 2357
AGS_TYPE_OPEN_SFZ_FILE, 2361
AGS_TYPE_OPEN_SINGLE_FILE, 2350
AGS_TYPE_OPEN_WAVE, 2365
AGS_TYPE_OSC_ACTION_CONTROLLER, 1721
AGS_TYPE_OSC_BUFFER_UTIL, 1645
AGS_TYPE_OSC_BUILDER, 1661
AGS_TYPE_OSC_CLIENT, 1683
AGS_TYPE_OSC_CONFIG_CONTROLLER, 1724
AGS_TYPE_OSC_CONNECTION, 1704
AGS_TYPE_OSC_CONTROLLER, 1718
AGS_TYPE_OSC_EXPORT_CONTROLLER, 1728
AGS_TYPE_OSC_FRONT_CONTROLLER, 1733
AGS_TYPE_OSC_INFO_CONTROLLER, 1737

AGS_TYPE_OSC_MESSAGE, 1709
AGS_TYPE_OSC_METER_CONTROLLER, 1745
AGS_TYPE_OSC_NODE_CONTROLLER, 1750
AGS_TYPE_OSC_PARSER, 1672
AGS_TYPE_OSC_PLUGIN_CONTROLLER, 1752
AGS_TYPE_OSC_RENEW_CONTROLLER, 1755
AGS_TYPE_OSC_RESPONSE, 1715
AGS_TYPE_OSC_SERVER, 1695
AGS_TYPE_OSC_STATUS_CONTROLLER, 1758
AGS_TYPE_OSC_UTIL, 1651
AGS_TYPE_OSC_WEBSOCKET_CONNECTION, 1766
AGS_TYPE_OSC_XMLRPC_CONTROLLER, 1775
AGS_TYPE_OSC_XMLRPC_MESSAGE, 1769
AGS_TYPE_OSC_XMLRPC_SERVER, 1763
AGS_TYPE_OSS_DEVIN, 680
AGS_TYPE_OSS_DEVOUT, 690
AGS_TYPE_OSS_MIDIIN, 700
AGS_TYPE_OUTPUT, 1099
AGS_TYPE_PATTERN, 1301
AGS_TYPE_PEAK_UTIL, 352
AGS_TYPE_PHASE_SHIFT_UTIL, 256
AGS_TYPE_PLAYBACK, 1334
AGS_TYPE_PLAYBACK_DOMAIN, 1327
AGS_TYPE_PLUGIN_PORT, 36
AGS_TYPE_PORT, 1157
AGS_TYPE_PRESET, 1315
AGS_TYPE_PULSE_CLIENT, 783
AGS_TYPE_PULSE_DEVIN, 799
AGS_TYPE_PULSE_DEVOUT, 809
AGS_TYPE_PULSE_PORT, 791
AGS_TYPE_PULSE_SERVER, 775
AGS_TYPE_RECALL, 1404
AGS_TYPE_RECALL_AUDIO, 1419
AGS_TYPE_RECALL_AUDIO_RUN, 1423
AGS_TYPE_RECALL_AUDIO_SIGNAL, 1443
AGS_TYPE_RECALL_CHANNEL, 1428
AGS_TYPE_RECALL_CHANNEL_RUN, 1435
AGS_TYPE_RECALL_CONTAINER, 1358
AGS_TYPE_RECALL_DEPENDENCY, 1363
AGS_TYPE_RECALL_FLAGS, 1404
AGS_TYPE_RECALL_ID, 1342
AGS_TYPE_RECALL_NOTIFY_DEPENDENCY_MODE, 1404
AGS_TYPE_RECALL_RECYCLING, 1439
AGS_TYPE_RECYCLING, 1117
AGS_TYPE_RECYCLING_CONTEXT, 1350
AGS_TYPE_REMOVE_AUDIO, 2368
AGS_TYPE_REMOVE_AUDIO_SIGNAL, 2371
AGS_TYPE_REMOVE_NOTE, 2374
AGS_TYPE_REMOVE_SOUNDCARD, 2377
AGS_TYPE_RESAMPLE_UTIL, 375
AGS_TYPE_RESET_FX_ANALYSE, 2386
AGS_TYPE_RESET_FX_PEAK, 2383
AGS_TYPE_RESET_NOTE, 2380
AGS_TYPE_RESIZE_AUDIO, 2388
AGS_TYPE_SEEK_SOUNDCARD, 2392
AGS_TYPE_SEQUENCER_THREAD, 1480
AGS_TYPE_SEQUENCER_UTIL, 161
AGS_TYPE_SET_AUDIO_CHANNELS, 2395
AGS_TYPE_SET_BUFFER_SIZE, 2398
AGS_TYPE_SET_DEVICE, 2401
AGS_TYPE_SET_FORMAT, 2403
AGS_TYPE_SET_MUTED, 2406
AGS_TYPE_SET_SAMPLERATE, 2409
AGS_TYPE_SF2_LOADER, 1492
AGS_TYPE_SF2_MIDI_LOCALE_LOADER, 1502
AGS_TYPE_SF2_SYNTH_GENERATOR, 531
AGS_TYPE_SF2_SYNTH_UTIL, 448
AGS_TYPE_SFZ_FILE, 1869
AGS_TYPE_SFZ_GROUP, 1876
AGS_TYPE_SFZ_INSTRUMENT_LOADER, 1518
AGS_TYPE_SFZ_LOADER, 1510
AGS_TYPE_SFZ_REGION, 1883
AGS_TYPE_SFZ_SAMPLE, 1892
AGS_TYPE_SFZ_SYNTH_GENERATOR, 547
AGS_TYPE_SFZ_SYNTH_UTIL, 465
AGS_TYPE_SNDFILE, 1861
AGS_TYPE_SOUND_ABILITY_FLAGS, 156
AGS_TYPE_SOUND_BEHAVIOUR_FLAGS, 156
AGS_TYPE_SOUND_CONTAINER, 1901
AGS_TYPE_SOUND_KEY_FORMAT, 156
AGS_TYPE_SOUND_PROVIDER, 954
AGS_TYPE_SOUND_RESOURCE, 1909
AGS_TYPE_SOUND_SCOPE, 156
AGS_TYPE_SOUND_STAGING_FLAGS, 156
AGS_TYPE_SOUND_STATE_FLAGS, 156
AGS_TYPE_SOUNDCARD_THREAD, 1484
AGS_TYPE_SOUNDCARD_UTIL, 160
AGS_TYPE_START_AUDIO, 2412
AGS_TYPE_START_CHANNEL, 2415
AGS_TYPE_START_SEQUENCER, 2418
AGS_TYPE_START_SOUNDCARD, 2420
AGS_TYPE_STOP_SEQUENCER, 2422
AGS_TYPE_STOP_SOUNDCARD, 2425
AGS_TYPE_STOP_THREAD, 2427
AGS_TYPE_SWITCH_BUFFER_FLAG, 2429
AGS_TYPE_SYNTH_GENERATOR, 508
AGS_TYPE_SYNTH_OSCILLATOR_MODE, 158
AGS_TYPE_SYNTH_UTIL, 401
AGS_TYPE_TIC_DEVICE, 2431
AGS_TYPE_TIME_STRETCH_UTIL, 245
AGS_TYPE_TOGGLE_PATTERN_BIT, 2434
AGS_TYPE_TRACK, 1291
AGS_TYPE_VOLUME_UTIL, 331
AGS_TYPE_WASAPI_DEVIN, 717
AGS_TYPE_WASAPI_DEVOUT, 707
AGS_TYPE_WAVE, 1262
AGS_TYPE_WAVE_LOADER, 1525

V

AGS_VOLUME_UTIL, 331
ags_volume_util_alloc, 323
ags_volume_util_compute, 330
ags_volume_util_compute_complex, 330

ags_volume_util_compute_double, 330
ags_volume_util_compute_float, 330
ags_volume_util_compute_s16, 329
ags_volume_util_compute_s24, 329
ags_volume_util_compute_s32, 329
ags_volume_util_compute_s64, 329
ags_volume_util_compute_s8, 328
ags_volume_util_copy, 323
AGS_VOLUME_UTIL_DEFAULT_FORMAT, 331
ags_volume_util_free, 324
ags_volume_util_get_buffer_length, 327
ags_volume_util_get_destination, 324
ags_volume_util_get_destination_stride, 325
ags_volume_util_get_format, 327
ags_volume_util_get_source, 325
ags_volume_util_get_source_stride, 326
ags_volume_util_get_type, 331
ags_volume_util_get_volume, 328
ags_volume_util_set_buffer_length, 327
ags_volume_util_set_destination, 324
ags_volume_util_set_destination_stride, 325
ags_volume_util_set_format, 328
ags_volume_util_set_source, 326
ags_volume_util_set_source_stride, 326
ags_volume_util_set_volume, 328
AgsVolumeUtil, 331

W

AGS_WASAPI_DEVIN, 714
ags_wasapi_devin_adjust_delay_and_attack, 713
AGS_WASAPI_DEVIN_CLASS, 714
AGS_WASAPI_DEVIN_DEFAULT_APP_BUFFER_SIZE, 715
AGS_WASAPI_DEVIN_DEFAULT_WASAPI_BUFFER_SIZE, 715
AGS_WASAPI_DEVIN_ERROR, 716
ags_wasapi_devin_error_quark, 712
ags_wasapi_devin_flags_get_type, 715
AGS_WASAPI_DEVIN_GET_CLASS, 714
AGS_WASAPI_DEVIN_GET_OBJ_MUTEX, 712
ags_wasapi_devin_get_type, 715
ags_wasapi_devin_new, 714
ags_wasapi_devin_realloc_buffer, 714
ags_wasapi_devin_set_flags, 712
ags_wasapi_devin_switch_buffer_flag, 713
ags_wasapi_devin_test_flags, 712
ags_wasapi_devin_unset_flags, 713
AGS_WASAPI_DEVOUT, 705
ags_wasapi_devout_adjust_delay_and_attack, 704
AGS_WASAPI_DEVOUT_CLASS, 705
AGS_WASAPI_DEVOUT_DEFAULT_APP_BUFFER_SIZE, 706
AGS_WASAPI_DEVOUT_DEFAULT_WASAPI_BUFFER_SIZE, 706
AGS_WASAPI_DEVOUT_ERROR, 707
ags_wasapi_devout_error_quark, 703
ags_wasapi_devout_flags_get_type, 705
AGS_WASAPI_DEVOUT_GET_CLASS, 705
AGS_WASAPI_DEVOUT_GET_OBJ_MUTEX, 703
ags_wasapi_devout_get_type, 706
ags_wasapi_devout_new, 705
ags_wasapi_devout_realloc_buffer, 704
ags_wasapi_devout_set_flags, 703
ags_wasapi_devout_switch_buffer_flag, 704
ags_wasapi_devout_test_flags, 703
ags_wasapi_devout_unset_flags, 704
AgsWasapiDevin, 717
AgsWasapiDevin:attack, 717
AgsWasapiDevin:bpm, 717
AgsWasapiDevin:buffer, 718
AgsWasapiDevin:buffer-size, 718
AgsWasapiDevin:delay-factor, 718
AgsWasapiDevin:device, 718
AgsWasapiDevin:dsp-channels, 719
AgsWasapiDevin:format, 719
AgsWasapiDevin:pcm-channels, 719
AgsWasapiDevin:samplerate, 719
AgsWasapiDevinAppBufferMode, 716
AgsWasapiDevinClass, 717
AgsWasapiDevinError, 717
AgsWasapiDevinFlags, 715
AgsWasapiDevout, 708
AgsWasapiDevout:attack, 708
AgsWasapiDevout:bpm, 708
AgsWasapiDevout:buffer, 708
AgsWasapiDevout:buffer-size, 709
AgsWasapiDevout:delay-factor, 709
AgsWasapiDevout:device, 709
AgsWasapiDevout:dsp-channels, 709
AgsWasapiDevout:format, 710
AgsWasapiDevout:pcm-channels, 710
AgsWasapiDevout:samplerate, 710
AgsWasapiDevoutAppBufferMode, 707
AgsWasapiDevoutClass, 708
AgsWasapiDevoutError, 707
AgsWasapiDevoutFlags, 706
AGS_WAVE, 1260
AgsWave, 1262
AgsWave:audio, 1262
AgsWave:buffer, 1262
AgsWave:buffer-size, 1263
AgsWave:format, 1263
AgsWave:line, 1263
AgsWave:samplerate, 1263
AgsWave:timestamp, 1263
ags_wave_add, 1253
ags_wave_add_all_to_selection, 1257
ags_wave_add_buffer, 1254
ags_wave_add_region_to_selection, 1257
AGS_WAVE_CLASS, 1260
AGS_WAVE_CLIPBOARD_FORMAT, 1262
AGS_WAVE_CLIPBOARD_TYPE, 1261
AGS_WAVE_CLIPBOARD_VERSION, 1261
ags_wave_copy_selection, 1258

ags_wave_cut_selection, 1258
AGS_WAVE_DEFAULT_BPM, 1261
AGS_WAVE_DEFAULT_BUFFER_LENGTH, 1261
AGS_WAVE_DEFAULT_DURATION, 1261
AGS_WAVE_DEFAULT_JIFFIE, 1261
AGS_WAVE_DEFAULT_LENGTH, 1261
AGS_WAVE_DEFAULT_OFFSET, 1261
ags_wave_find_exact_point, 1255
ags_wave_find_near_timestamp, 1248
ags_wave_find_point, 1255
ags_wave_find_region, 1256
ags_wave_flags_get_type, 1260
ags_wave_free_all_selection, 1256
ags_wave_free_selection, 1256
ags_wave_get_audio, 1249
ags_wave_get_buffer, 1253
ags_wave_get_buffer_size, 1251
AGS_WAVE_GET_CLASS, 1260
ags_wave_get_format, 1251
ags_wave_get_line, 1249
AGS_WAVE_GET_OBJ_MUTEX, 1246
ags_wave_get_obj_mutex, 1247
ags_wave_get_samplerate, 1250
ags_wave_get_selection, 1254
ags_wave_get_timestamp, 1252
ags_wave_get_type, 1260
ags_wave_insert_from_clipboard, 1258
ags_wave_insert_from_clipboard_extended, 1259
ags_wave_is_buffer_selected, 1255
AGS_WAVE_LOADER, 1525
AGS_WAVE_LOADER_CLASS, 1525
ags_wave_loader_get_audio, 1522
ags_wave_loader_get_audio_file, 1523
AGS_WAVE_LOADER_GET_CLASS, 1525
ags_wave_loader_get_filename, 1523
AGS_WAVE_LOADER_GET_OBJ_MUTEX, 1521
ags_wave_loader_get_type, 1525
ags_wave_loader_new, 1524
ags_wave_loader_set_audio, 1522
ags_wave_loader_set_audio_file, 1524
ags_wave_loader_set_filename, 1523
ags_wave_loader_set_flags, 1521
ags_wave_loader_start, 1524
ags_wave_loader_test_flags, 1521
ags_wave_loader_unset_flags, 1522
AGS_WAVE_MINIMUM_BUFFER_LENGTH, 1261
ags_wave_new, 1259
ags_wave_remove_buffer, 1254
ags_wave_remove_region_from_selection, 1257
ags_wave_set_audio, 1249
ags_wave_set_buffer, 1253
ags_wave_set_buffer_size, 1251
ags_wave_set_flags, 1247
ags_wave_set_format, 1252
ags_wave_set_line, 1250
ags_wave_set_samplerate, 1250
ags_wave_set_timestamp, 1252
ags_wave_sort_func, 1248
ags_wave_test_flags, 1247
AGS_WAVE_TICS_PER_BEAT, 1261
ags_wave_unset_flags, 1248
AgsWaveClass, 1262
AgsWaveFlags, 1262
AgsWaveLoader, 1526
AgsWaveLoader:audio, 1526
AgsWaveLoader:audio-file, 1526
AgsWaveLoader:filename, 1526
AgsWaveLoaderClass, 1526
AgsWaveLoaderFlags, 1525