Advanced Gtk+ Sequencer Reference Manual

Mr. Joël Krähemann

Copyright (C)

COLLABORATORS

	TITLE : Advanced Gtk+ Sequencer Reference Manual	

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

I Library - Completion routines 1

1 AgsBufferUtil 2

2 AgsComplex 17

3 AgsConversion 21

4 AgsEndian 26

5 AgsMathUtil 29

6 AgsFunction 40

7 AgsSolverMatrix 45

8 AgsSolverVector 51

9 AgsSolverPolynomial 56

10 AgsLog 67

11 AgsRegex 71

12 AgsStringUtil 73

13 AgsTime 77

14 AgsTurtle 80

15 AgsTurtleManager 112

16 AgsUUID 116

II Util - Helper functions 120

17 AgsDestroyUtil 121

18 AgsIdGenerator 123

19 AgsListUtil 124

20 AgsSoundcardHelper 126

III Object - Interfaces and marshallers 129

21 AgsApplicable 130

22 AgsApplicationContext 134

23 AgsConfig 145

24 AgsPriority 154

25 AgsConnectable 162

26 AgsCountable 170

27 AgsCursor 174

28 ags_globals 181

29 ags_macros 183

30 AgsMainLoop 184

31 ags_marshal 191

32 AgsMutable 211

33 AgsPlugin 214

34 AgsPortlet 221

35 AgsSeekable 225

36 AgsSequencer 230

37 AgsSoundcard 245

38 AgsSoundServer 271

39 AgsTactable 278

IV XML Input/Output - Persist project 288

40 AgsFile 289

41 AgsFileIdRef 314

42 AgsFileLink 319

43 AgsFileLookup 324

44 AgsFileLaunch 329

V Thread - Multi-threaded tree 334

45 AgsConcurrencyProvider 335

46 AgsDestroyWorker 341

47 AgsGenericMainLoop 346

48 AgsMessageDelivery 349

49 AgsMessageQueue 357

50 AgsMessageEnvelope 365

51 AgsReturnableThread 371

52 AgsTaskLauncher 378

53 AgsTask 386

54 AgsTaskCompletion 392

55 AgsThread 399

56 AgsThreadApplicationContext 427

57 AgsThreadPool 431

58 AgsTimestamp 437

59 AgsWorkerThread 443

VI XML-RPC - server 449

60 AgsRegistry 450

61 AgsServer 455

62 AgsServerApplicationContext 467

63 AgsServerStatus **470**

64 AgsServiceProvider **473**

65 Controller - providing context path **478**

65.1 AgsController 478
65.2 AgsFrontController 484
65.3 AgsPluginController 488

66 Security - basic authentication **491**

66.1 AgsAuthentication 491
66.2 AgsAuthenticationManager 496
66.3 AgsBusinessGroup 505
66.4 AgsBusinessGroupManager 510
66.5 AgsCertificate 514
66.6 AgsCertificateManager 522
66.7 AgsPasswordStore 526
66.8 AgsPasswordStoreManager 530
66.9 AgsAuthSecurityContext 535
66.10 AgsSecurityContext 537
66.11 AgsXmlAuthentication 543
66.12 AgsXmlBusinessGroup 546
66.13 AgsXmlCertificate 548
66.14 AgsXmlPasswordStore 551

VII Annotation Glossary **555**

VIII Tree Index **558**

67 Index **561**

Part I

Library - Completion routines

Chapter 1

AgsBufferUtil

AgsBufferUtil — buffer util

Stability Level

Stable, unless otherwise indicated

Functions

guchar *	ags_buffer_util_s8_to_char_buffer ()
guchar *	ags_buffer_util_s16_to_char_buffer ()
guchar *	ags_buffer_util_s24_to_char_buffer ()
guchar *	ags_buffer_util_s32_to_char_buffer ()
guchar *	ags_buffer_util_s64_to_char_buffer ()
guchar *	ags_buffer_util_float_to_char_buffer ()
guchar *	ags_buffer_util_double_to_char_buffer ()
guchar *	ags_buffer_util_complex_to_char_buffer ()
gint8 *	ags_buffer_util_char_buffer_to_s8 ()
gint16 *	ags_buffer_util_char_buffer_to_s16 ()
gint32 *	ags_buffer_util_char_buffer_to_s24 ()
gint32 *	ags_buffer_util_char_buffer_to_s32 ()
gint64 *	ags_buffer_util_char_buffer_to_s64 ()
gfloat *	ags_buffer_util_char_buffer_to_float ()
gdouble *	ags_buffer_util_char_buffer_to_double ()
AgsComplex *	ags_buffer_util_char_buffer_to_complex ()
gint8	ags_buffer_util_char_buffer_read_s8 ()
gint16	ags_buffer_util_char_buffer_read_s16 ()
gint32	ags_buffer_util_char_buffer_read_s24 ()
gint32	ags_buffer_util_char_buffer_read_s32 ()
gint64	ags_buffer_util_char_buffer_read_s64 ()
gfloat	ags_buffer_util_char_buffer_read_float ()
gdouble	ags_buffer_util_char_buffer_read_double ()
AgsComplex *	ags_buffer_util_char_buffer_read_complex ()
void	ags_buffer_util_char_buffer_write_s8 ()
void	ags_buffer_util_char_buffer_write_s16 ()
void	ags_buffer_util_char_buffer_write_s24 ()
void	ags_buffer_util_char_buffer_write_s32 ()

void	ags_buffer_util_char_buffer_write_s64 ()
void	ags_buffer_util_char_buffer_write_float ()
void	ags_buffer_util_char_buffer_write_double ()
void	ags_buffer_util_char_buffer_write_complex ()
void	ags_buffer_util_char_buffer_swap_bytes ()
GType	ags_buffer_util_get_type ()

Types and Values

#define	AGS_TYPE_BUFFER_UTIL
struct	AgsBufferUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsBufferUtil
```

Includes

```
#include <ags/lib/ags_buffer_util.h>
```

Description

Common utility functions related to char buffers.

Functions

ags_buffer_util_s8_to_char_buffer ()

```
guchar~*
ags_buffer_util_s8_to_char_buffer (gint8 *buffer,
                                   guint buffer_length);
```

Pack *buffer* into an guchar buffer.

Parameters

buffer	the gint8 buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_s16_to_char_buffer ()

```
guchar~*
ags_buffer_util_s16_to_char_buffer (gint16 *buffer,
                                    guint buffer_length);
```

Pack *buffer* into an guchar buffer.

Parameters

buffer	the gint16 buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_s24_to_char_buffer ()

```
guchar~*
ags_buffer_util_s24_to_char_buffer (gint32 *buffer,
                                    guint buffer_length);
```

Pack *buffer* into an guchar buffer.

Parameters

buffer	the gint32 buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_s32_to_char_buffer ()

```
guchar~*
ags_buffer_util_s32_to_char_buffer (gint32 *buffer,
                                    guint buffer_length);
```

Pack *buffer* into an guchar buffer.

Parameters

buffer	the gint32 buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_s64_to_char_buffer ()

```
guchar~*
ags_buffer_util_s64_to_char_buffer (gint64 *buffer,
                                    guint buffer_length);
```

Pack `buffer` into an guchar buffer.

Parameters

buffer	the gint64 buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_float_to_char_buffer ()

```
guchar~*
ags_buffer_util_float_to_char_buffer (gfloat *buffer,
                                      guint buffer_length);
```

Pack `buffer` into an guchar buffer.

Parameters

buffer	the gfloat buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_double_to_char_buffer ()

```
guchar~*
ags_buffer_util_double_to_char_buffer (gdouble *buffer,
                                       guint buffer_length);
```

Pack *buffer* into an guchar buffer.

Parameters

buffer	the gdouble buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_complex_to_char_buffer ()

```
guchar~*
ags_buffer_util_complex_to_char_buffer
                                (AgsComplex *buffer,
                                 guint buffer_length);
```

Pack *buffer* into an guchar buffer.

Parameters

buffer	the AgsComplex buffer	
buffer_length	the buffer length	

Returns

the guchar buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_s8 ()

```
gint8~*
ags_buffer_util_char_buffer_to_s8 (guchar *cbuffer,
                                   guint buffer_size);
```

Unpack *cbuffer* to a gint8 buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gint8 buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_s16 ()

```
gint16~*
ags_buffer_util_char_buffer_to_s16 (guchar *cbuffer,
                                    guint buffer_size);
```

Unpack `cbuffer` to a gint16 buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gint16 buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_s24 ()

```
gint32~*
ags_buffer_util_char_buffer_to_s24 (guchar *cbuffer,
                                    guint buffer_size);
```

Unpack `cbuffer` to a gint32 buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gint32 buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_s32 ()

```
gint32~*
ags_buffer_util_char_buffer_to_s32 (guchar *cbuffer,
                                    guint buffer_size);
```

Unpack *cbuffer* to a gint32 buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gint32 buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_s64 ()

```
gint64~*
ags_buffer_util_char_buffer_to_s64 (guchar *cbuffer,
                                    guint buffer_size);
```

Unpack *cbuffer* to a gint64 buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gint64 buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_float ()

```
gfloat~*
ags_buffer_util_char_buffer_to_float (guchar *cbuffer,
                                      guint buffer_size);
```

Unpack *cbuffer* to a gfloat buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gfloat buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_double ()

```
gdouble~*
ags_buffer_util_char_buffer_to_double (guchar *cbuffer,
                                       guint buffer_size);
```

Unpack `cbuffer` to a gdouble buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the gdouble buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_to_complex ()

```
AgsComplex~*
ags_buffer_util_char_buffer_to_complex
                                (guchar *cbuffer,
                                 guint buffer_size);
```

Unpack `cbuffer` to a AgsComplex buffer

Parameters

cbuffer	the guchar buffer	
buffer_size	the buffer size	

Returns

the AgsComplex buffer.

[transfer full]

Since: 3.0.0

ags_buffer_util_char_buffer_read_s8 ()

```
gint8
ags_buffer_util_char_buffer_read_s8 (guchar *cbuffer,
                                     guint byte_order);
```

Read a gint8 quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order, ignored here	

Returns

the gint8 value

Since: 3.0.0

ags_buffer_util_char_buffer_read_s16 ()

```
gint16
ags_buffer_util_char_buffer_read_s16 (guchar *cbuffer,
                                      guint byte_order);
```

Read a gint16 quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the gint16 value

Since: 3.0.0

ags_buffer_util_char_buffer_read_s24 ()

```
gint32
ags_buffer_util_char_buffer_read_s24 (guchar *cbuffer,
                                      guint byte_order);
```

Read a gint32 24 bit quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the gint32 value

Since: 3.0.0

ags_buffer_util_char_buffer_read_s32 ()

```
gint32
ags_buffer_util_char_buffer_read_s32 (guchar *cbuffer,
                                      guint byte_order);
```

Read a gint32 32 bit quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the gint32 value

Since: 3.0.0

ags_buffer_util_char_buffer_read_s64 ()

```
gint64
ags_buffer_util_char_buffer_read_s64 (guchar *cbuffer,
                                      guint byte_order);
```

Read a gint64 64 bit quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the gint64 value

Since: 3.0.0

ags_buffer_util_char_buffer_read_float ()

```
gfloat
ags_buffer_util_char_buffer_read_float
                                (guchar *cbuffer,
                                 guint byte_order);
```

Read a gfloat quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the gfloat value

Since: 3.0.0

ags_buffer_util_char_buffer_read_double ()

```
gdouble
ags_buffer_util_char_buffer_read_double
                                (guchar *cbuffer,
                                 guint byte_order);
```

Read a gdouble quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the gdouble value

Since: 3.0.0

ags_buffer_util_char_buffer_read_complex ()

```
AgsComplex~*
ags_buffer_util_char_buffer_read_complex
                                (guchar *cbuffer,
                                 guint byte_order);
```

Read a AgsComplex quantity of *cbuffer* .

Parameters

cbuffer	the character buffer	
byte_order	reverse order	

Returns

the AgsComplex value

Since: 3.0.0

ags_buffer_util_char_buffer_write_s8 ()

```
void
ags_buffer_util_char_buffer_write_s8 (guchar *cbuffer,
                                      gint8 value,
                                      guint byte_order);
```

Write a gint8 quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gint8 value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_s16 ()

```
void
ags_buffer_util_char_buffer_write_s16 (guchar *cbuffer,
                                       gint16 value,
                                       guint byte_order);
```

Write a gint16 quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gint16 value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_s24 ()

```
void
ags_buffer_util_char_buffer_write_s24 (guchar *cbuffer,
                                       gint32 value,
                                       guint byte_order);
```

Write a gint32 quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gint32 value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_s32 ()

```
void
ags_buffer_util_char_buffer_write_s32 (guchar *cbuffer,
                                       gint32 value,
                                       guint byte_order);
```

Write a gint32 quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gint32 value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_s64 ()

```
void
ags_buffer_util_char_buffer_write_s64 (guchar *cbuffer,
                                       gint64 value,
                                       guint byte_order);
```

Write a gint64 quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gint64 value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_float ()

```
void
ags_buffer_util_char_buffer_write_float
                                (guchar *cbuffer,
                                 gfloat value,
                                 guint byte_order);
```

Write a gfloat quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gfloat value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_double ()

```
void
ags_buffer_util_char_buffer_write_double
                                (guchar *cbuffer,
                                 gdouble value,
                                 guint byte_order);
```

Write a gdouble quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the gdouble value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_write_complex ()

```
void
ags_buffer_util_char_buffer_write_complex
                                (guchar *cbuffer,
                                 AgsComplex *value,
                                 guint byte_order);
```

Write a AgsComplex quantity to *cbuffer* .

Parameters

cbuffer	the character buffer	
value	the AgsComplex value	
byte_order	reverse order	

Since: 3.0.0

ags_buffer_util_char_buffer_swap_bytes ()

```
void
ags_buffer_util_char_buffer_swap_bytes
                                (guchar *cbuffer,
                                 guint word_size,
                                 guint buffer_size);
```

Swap bytes in view of Little/Big-Endian.

Parameters

cbuffer	the buffer to swap	
word_size	the word size	
buffer_size	the buffer size	

Since: 3.0.0

ags_buffer_util_get_type ()

```
GType
ags_buffer_util_get_type (void);
```

Types and Values

AGS_TYPE_BUFFER_UTIL

```
#define AGS_TYPE_BUFFER_UTIL         (ags_buffer_util_get_type())
```

struct AgsBufferUtil

```
struct AgsBufferUtil {
  //empty
};
```

Chapter 2

AgsComplex

AgsComplex — Boxed type of complex

Stability Level

Stable, unless otherwise indicated

Functions

AgsComplex *	ags_complex_alloc ()
gpointer	ags_complex_copy ()
void	ags_complex_free ()
void	ags_complex_set ()
void	ags_complex_get_term ()
void	ags_complex_set_term ()
#define	AGS_COMPLEX()
GType	ags_complex_get_type ()

Types and Values

#define	AGS_COMPLEX_M_E
#define	AGS_COMPLEX_M_PI
#define	AGS_TYPE_COMPLEX
struct	AgsComplex

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsComplex
```

Includes

```
#include <ags/lib/ags_complex.h>
```

Description

Boxed type of complex data type.

Functions

ags_complex_alloc ()

```
AgsComplex~*
ags_complex_alloc ();
```

Allocate AgsComplex

Returns

a new AgsComplex

Since: 3.0.0

ags_complex_copy ()

```
gpointer
ags_complex_copy (AgsComplex *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsComplex

Returns

a pointer of the new AgsComplex

Since: 3.0.0

ags_complex_free ()

```
void
ags_complex_free (AgsComplex *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsComplex

Since: 3.0.0

ags_complex_set ()

```
void
ags_complex_set (AgsComplex *ptr);
```

Set complex number.

Parameters

ptr	the AgsComplex	
real	the real part	
imag	the imaginary part	

Since: 3.7.11

ags_complex_get_term ()

```
void
ags_complex_get_term (AgsComplex *ptr,
                      gdouble *real,
                      gdouble *imag);
```

ags_complex_set_term ()

```
void
ags_complex_set_term (AgsComplex *ptr,
                      gdouble real,
                      gdouble imag);
```

AGS_COMPLEX()

```
#define AGS_COMPLEX(ptr) ((AgsComplex *)(ptr))
```

ags_complex_get_type ()

```
GType
ags_complex_get_type (void);
```

Types and Values

AGS_COMPLEX_M_E

```
#define AGS_COMPLEX_M_E (M_E + I * 0.0)
```

AGS_COMPLEX_M_PI

```
#define AGS_COMPLEX_M_PI (M_PI + I * 0.0)
```

AGS_TYPE_COMPLEX

```
#define AGS_TYPE_COMPLEX                (ags_complex_get_type())
```

struct AgsComplex

```
struct AgsComplex {
  gdouble real;
  gdouble imag;
};
```

Chapter 3

AgsConversion

AgsConversion — abstract conversion

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CONVERSION_GET_OBJ_MUTEX()
gdouble	ags_conversion_convert ()
AgsConversion *	ags_conversion_new ()
#define	AGS_CONVERSION()
#define	AGS_CONVERSION_CLASS()
#define	AGS_CONVERSION_GET_CLASS()
#define	AGS_IS_CONVERSION()
#define	AGS_IS_CONVERSION_CLASS()
GType	ags_conversion_get_type ()

Properties

char *	description	Read / Write
char *	name	Read / Write

Signals

double	convert	Run Last

Types and Values

#define	AGS_TYPE_CONVERSION
struct	AgsConversion
struct	AgsConversionClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsConversion
```

Includes

```
#include <ags/lib/ags_conversion.h>
```

Description

AgsConversion does a zero conversion it returns during ::convert() the very same value.

Functions

AGS_CONVERSION_GET_OBJ_MUTEX()

```
#define AGS_CONVERSION_GET_OBJ_MUTEX(obj) (&(((AgsConversion *) obj)->obj_mutex))
```

ags_conversion_convert ()

```
gdouble
ags_conversion_convert (AgsConversion *conversion,
                        gdouble x,
                        gboolean reverse);
```

Convert a value if *reverse* is FALSE then use the target format otherwise if TRUE convert back to original format.

Parameters

conversion	the AgsConversion	
x	the value to convert	
reverse	the direction to convert	

Returns

the converted value as gdouble

Since: 3.0.0

ags_conversion_new ()

```
AgsConversion~*
ags_conversion_new ();
```

Instantiate a new AgsConversion.

Returns

the new instance

Since: 3.0.0

AGS_CONVERSION()

```
#define AGS_CONVERSION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CONVERSION, AgsConversion))
```

AGS_CONVERSION_CLASS()

```
#define AGS_CONVERSION_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CONVERSION, AgsConversionClass))
```

AGS_CONVERSION_GET_CLASS()

```
#define AGS_CONVERSION_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_CONVERSION, AgsConversionClass))
```

AGS_IS_CONVERSION()

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

AGS_IS_CONVERSION_CLASS()

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

ags_conversion_get_type ()

```
GType
ags_conversion_get_type (void);
```

Types and Values

AGS_TYPE_CONVERSION

```
#define AGS_TYPE_CONVERSION                 (ags_conversion_get_type())
```

struct AgsConversion

```
struct AgsConversion;
```

struct AgsConversionClass

```
struct AgsConversionClass {
  GObjectClass gobject;

  gdouble (*convert)(AgsConversion *conversion,
         gdouble x,
         gboolean reverse);
};
```

Property Details

The "description" property

```
  "description"              char~*
```

The description of the conversion.

Owner: AgsConversion

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "name" property

```
  "name"                     char~*
```

The name of the conversion.

Owner: AgsConversion

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Signal Details

The "convert" signal

```
double
user_function (AgsConversion *conversion,
               double         x,
               gboolean       reverse,
               gpointer       user_data)
```

The ::convert signal notifies about converted value.

Parameters

conversion	the AgsConversion to modify	
x	the value to convert	
reverse	the direction to translate	
user_data	user data set when the signal handler was connected.	

Returns

the converted value

Flags: Run Last

Since: 3.0.0

Chapter 4

AgsEndian

AgsEndian — dealing with endiannes

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_endian_host_is_le ()
gboolean	ags_endian_host_is_be ()
gfloat	ags_endian_swap_float ()
GType	ags_byte_order_get_type ()

Types and Values

enum	AgsByteOrder
#define	AGS_TYPE_BYTE_ORDER

Object Hierarchy

```
    GEnum
    &#x2570;&#x2500;&#x2500; AgsByteOrder
```

Includes

```
#include <ags/lib/ags_endian.h>
```

Description

Some common routines dealing with endiannes.

Functions

ags_endian_host_is_le ()

```
gboolean
ags_endian_host_is_le ();
```

Check host is Little Endian.

Returns

TRUE if LE, otherwise FALSE

Since: 3.0.0

ags_endian_host_is_be ()

```
gboolean
ags_endian_host_is_be ();
```

Check host is Big Endian.

Returns

TRUE if BE, otherwise FALSE

Since: 3.0.0

ags_endian_swap_float ()

```
gfloat
ags_endian_swap_float (gfloat x);
```

Swaps the floating point numbers endianness.

Parameters

x	the float to swap

Returns

the byte-swapped float

Since: 3.0.0

ags_byte_order_get_type ()

```
GType
ags_byte_order_get_type ();
```

Types and Values

enum AgsByteOrder

Endianess.

Members

AGS_BYTE_ORDER_LE	little endian
AGS_BYTE_ORDER_BE	big endian

AGS_TYPE_BYTE_ORDER

```
#define AGS_TYPE_BYTE_ORDER          (ags_byte_order_get_type())
```

Chapter 5

AgsMathUtil

AgsMathUtil — math util

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_math_util_find_parenthesis_all ()
void	ags_math_util_find_exponent_parenthesis ()
void	ags_math_util_find_function_parenthesis ()
void	ags_math_util_find_term_parenthesis ()
gboolean	ags_math_util_match_sign ()
gboolean	ags_math_util_match_coefficient ()
gboolean	ags_math_util_match_symbol ()
gboolean	ags_math_util_match_exponent ()
gboolean	ags_math_util_match_operator ()
gboolean	ags_math_util_match_function ()
gboolean	ags_math_util_coefficient_to_complex ()
AgsComplex *	ags_math_util_multiply_coefficient_all ()
gchar *	ags_math_util_find_function ()
gchar *	ags_math_util_find_symbol ()
gchar **	ags_math_util_find_symbol_all ()
gboolean	ags_math_util_is_term ()
void	ags_math_util_split_polynomial ()
void	ags_math_util_split_sum ()
GType	ags_math_util_get_type ()

Types and Values

#define	AGS_SYMBOLIC_EULER
#define	AGS_SYMBOLIC_PI
#define	AGS_SYMBOLIC_INFINIT
#define	AGS_SYMBOLIC_COMPLEX_UNIT

#define	AGS_SUBSCRIPT_0
#define	AGS_SUBSCRIPT_1
#define	AGS_SUBSCRIPT_2
#define	AGS_SUBSCRIPT_3
#define	AGS_SUBSCRIPT_4
#define	AGS_SUBSCRIPT_5
#define	AGS_SUBSCRIPT_6
#define	AGS_SUBSCRIPT_7
#define	AGS_SUBSCRIPT_8
#define	AGS_SUBSCRIPT_9
#define	AGS_TYPE_MATH_UTIL
struct	AgsMathUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsMathUtil
```

Includes

```
#include <ags/lib/ags_math_util.h>
```

Description

Common math utility functions.

Functions

ags_math_util_find_parenthesis_all ()

```
void
ags_math_util_find_parenthesis_all (gchar *str,
                                    gint **open_position,
                                    gint **close_position,
                                    guint *open_position_count,
                                    guint *close_position_count);
```

Find all parenthesis.

Parameters

str	the string	
open_position	open position array return location.	*[out]*
close_position	close position array return location.	*[out]*
open_position_count	open position count return location.	*[out]*

close_position_count	close position count return location.	*[out]*

Since: 3.2.0

ags_math_util_find_exponent_parenthesis ()

```
void
ags_math_util_find_exponent_parenthesis
                                (gchar *str,
                                 gint **exponent_open_position,
                                 gint **exponent_close_position,
                                 guint *exponent_open_position_count,
                                 guint *exponent_close_position_count);
```

Find exponent parenthesis.

Parameters

str	the string	
exponent_open_position	exponent open position array return location.	*[out]*
exponent_close_position	exponent close position array return location.	*[out]*
exponent_open_position_count	exponent open position count return location.	*[out]*
exponent_close_position_count	exponent close position count return location.	*[out]*

Since: 3.2.0

ags_math_util_find_function_parenthesis ()

```
void
ags_math_util_find_function_parenthesis
                                (gchar *str,
                                 gint **function_open_position,
                                 gint **function_close_position,
                                 guint *function_open_position_count,
                                 guint *function_close_position_count);
```

Find function parenthesis.

Parameters

str	the string	
function_open_position	function open position array return location.	*[out]*
function_close_position	function close position array return location.	*[out]*
function_open_position_count	function open position count return location.	*[out]*

function_close_position_count	function close position count return location.	*[out]*

Since: 3.2.0

ags_math_util_find_term_parenthesis ()

```
void
ags_math_util_find_term_parenthesis (gchar *str,
                                     gint **term_open_position,
                                     gint **term_close_position,
                                     guint *term_open_position_count,
                                     guint *term_close_position_count);
```

Find term parenthesis.

Parameters

str	the string	
term_open_position	term open position array return location.	*[out]*
term_close_position	term close position array return location.	*[out]*
term_open_position_count	term open position count return location.	*[out]*
term_close_position_count	term close position count return location.	*[out]*

Since: 3.2.0

ags_math_util_match_sign ()

```
gboolean
ags_math_util_match_sign (gchar *offset,
                          gchar *end_ptr,
                          gchar **start_offset,
                          gchar **end_offset);
```

Match sign.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_match_coefficient ()

```
gboolean
ags_math_util_match_coefficient (gchar *offset,
                                 gchar *end_ptr,
                                 gchar **start_offset,
                                 gchar **end_offset);
```

Match coefficient including optional sign.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_match_symbol ()

```
gboolean
ags_math_util_match_symbol (gchar *offset,
                            gchar *end_ptr,
                            gchar **start_offset,
                            gchar **end_offset);
```

Match symbol including optional sign.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_match_exponent ()

```
gboolean
ags_math_util_match_exponent (gchar *offset,
                              gchar *end_ptr,
                              gchar **start_offset,
                              gchar **end_offset);
```

Match exponent with or without parenthesis.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_match_operator ()

```
gboolean
ags_math_util_match_operator (gchar *offset,
                              gchar *end_ptr,
                              gchar **start_offset,
                              gchar **end_offset);
```

Match operator.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_match_function ()

```
gboolean
ags_math_util_match_function (gchar *offset,
                              gchar *end_ptr,
                              gchar **start_offset,
                              gchar **end_offset);
```

Match function.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_coefficient_to_complex ()

```
gboolean
ags_math_util_coefficient_to_complex (gchar *coefficient,
                                      AgsComplex *value);
```

Compute *value* from *coefficient* .

Parameters

coefficient	the coefficient string	
value	return location of value.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.6.0

ags_math_util_multiply_coefficient_all ()

```
AgsComplex~*
ags_math_util_multiply_coefficient_all
                                (gchar **coefficient,
                                 guint *value_count);
```

ags_math_util_find_function ()

```
gchar~*
ags_math_util_find_function (gchar *str);
```

Find next function.

Parameters

str	the string

Returns

the string offset matching function, otherwise NULL

Since: 3.2.0

ags_math_util_find_symbol ()

```
gchar~*
ags_math_util_find_symbol (gchar *str);
```

Find next symbol.

Parameters

str	the string

Returns

the string offset matching symbol, otherwise NULL

Since: 3.2.0

ags_math_util_find_symbol_all ()

```
gchar~**
ags_math_util_find_symbol_all (gchar *str);
```

Find all symbols.

Parameters

str	the string

Returns

the string vector containing the symbols

Since: 3.2.0

ags_math_util_is_term ()

```
gboolean
ags_math_util_is_term (gchar *term);
```

Test if *term* is a term.

Parameters

term	the term	

Returns

TRUE on success, otherwise FALSE

Since: 3.2.0

ags_math_util_split_polynomial ()

```
void
ags_math_util_split_polynomial (gchar *polynomial,
                                gchar ***factor,
                                gchar ***factor_exponent);
```

Split *polynomial* into coefficient, powers of symbols and summand.

Parameters

polynomial	the polynomial	
factor	the return location of factors.	*[out]*
factor_exponent	the return location of factor exponents.	*[out]*

Since: 3.2.0

ags_math_util_split_sum ()

```
void
ags_math_util_split_sum (gchar *sum,
                         gchar ***summand);
```

Split *sum* into summands.

Parameters

sum	the sum	
summand	the return location of summands.	*[out]*

Since: 3.2.0

ags_math_util_get_type ()

```
GType
ags_math_util_get_type (void);
```

Types and Values

AGS_SYMBOLIC_EULER

```
#define AGS_SYMBOLIC_EULER "&#x212f;"
```

AGS_SYMBOLIC_PI

```
#define AGS_SYMBOLIC_PI "&#x1d70b;"
```

AGS_SYMBOLIC_INFINIT

```
#define AGS_SYMBOLIC_INFINIT "∞"
```

AGS_SYMBOLIC_COMPLEX_UNIT

```
#define AGS_SYMBOLIC_COMPLEX_UNIT "&#x1d456;"
```

AGS_SUBSCRIPT_0

```
#define AGS_SUBSCRIPT_0 "&#x2080;"
```

AGS_SUBSCRIPT_1

```
#define AGS_SUBSCRIPT_1 "&#x2081;"
```

AGS_SUBSCRIPT_2

```
#define AGS_SUBSCRIPT_2 "&#x2082;"
```

AGS_SUBSCRIPT_3

```
#define AGS_SUBSCRIPT_3 "&#x2083;"
```

AGS_SUBSCRIPT_4

```
#define AGS_SUBSCRIPT_4 "&#x2084;"
```

AGS_SUBSCRIPT_5

```
#define AGS_SUBSCRIPT_5 "&#x2085;"
```

AGS_SUBSCRIPT_6

```
#define AGS_SUBSCRIPT_6 "&#x2086;"
```

AGS_SUBSCRIPT_7

```
#define AGS_SUBSCRIPT_7 "&#x2087;"
```

AGS_SUBSCRIPT_8

```
#define AGS_SUBSCRIPT_8 "&#x2088;"
```

AGS_SUBSCRIPT_9

```
#define AGS_SUBSCRIPT_9 "&#x2089;"
```

AGS_TYPE_MATH_UTIL

```
#define AGS_TYPE_MATH_UTIL          (ags_math_util_get_type())
```

struct AgsMathUtil

```
struct AgsMathUtil {
  //empty
};
```

Chapter 6

AgsFunction

AgsFunction — Function to translate values

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FUNCTION_GET_OBJ_MUTEX()
void	ags_function_add_matrix ()
void	ags_function_remove_matrix ()
AgsFunction *	ags_function_new ()
#define	AGS_FUNCTION()
#define	AGS_FUNCTION_CLASS()
#define	AGS_FUNCTION_GET_CLASS()
#define	AGS_IS_FUNCTION()
#define	AGS_IS_FUNCTION_CLASS()
GType	ags_function_get_type ()

Properties

char *	normalized-function	Read / Write
gpointer	pivot-table	Read / Write
char *	source-function	Read / Write

Types and Values

#define	AGS_TYPE_FUNCTION
struct	AgsFunction
struct	AgsFunctionClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFunction
```

Includes

```
#include <ags/lib/ags_function.h>
```

Description

The AgsFunction translates values from linear math to a given function.

Functions

AGS_FUNCTION_GET_OBJ_MUTEX()

```
#define AGS_FUNCTION_GET_OBJ_MUTEX(obj) (&(((AgsFunction *) obj)->obj_mutex))
```

ags_function_add_matrix ()

```
void
ags_function_add_matrix (AgsFunction *function,
                         AgsSolverMatrix *solver_matrix);
```

Add *solver_matrix* to *function*.

Parameters

function	the AgsFunction	
solver_matrix	the AgsSolverMatrix	

Since: 3.2.0

ags_function_remove_matrix ()

```
void
ags_function_remove_matrix (AgsFunction *function,
                            AgsSolverMatrix *solver_matrix);
```

Remove *solver_matrix* from *function*.

Parameters

function	the AgsFunction	
solver_matrix	the AgsSolverMatrix	

Since: 3.2.0

ags_function_new ()

```
AgsFunction~*
ags_function_new (gchar *source_function);
```

Instantiate a new AgsFunction.

Parameters

source_function	the source function	

Returns

the new instance

Since: 3.0.0

AGS_FUNCTION()

```
#define AGS_FUNCTION(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FUNCTION, AgsFunction))
```

AGS_FUNCTION_CLASS()

```
#define AGS_FUNCTION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FUNCTION, AgsFunctionClass))
```

AGS_FUNCTION_GET_CLASS()

```
#define AGS_FUNCTION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (obj, AGS_TYPE_FUNCTION ↩
    , AgsFunctionClass))
```

AGS_IS_FUNCTION()

```
#define AGS_IS_FUNCTION(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FUNCTION))
```

AGS_IS_FUNCTION_CLASS()

```
#define AGS_IS_FUNCTION_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FUNCTION))
```

ags_function_get_type ()

```
GType
ags_function_get_type (void);
```

Types and Values

AGS_TYPE_FUNCTION

```
#define AGS_TYPE_FUNCTION                (ags_function_get_type())
```

struct AgsFunction

```
struct AgsFunction;
```

struct AgsFunctionClass

```
struct AgsFunctionClass {
  GObjectClass gobject;
};
```

Property Details

The "normalized-function" property

```
  "normalized-function"      char~*
```

The normalized function.

Owner: AgsFunction

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "pivot-table" property

```
  "pivot-table"              gpointer
```

The pivot table.

Owner: AgsFunction

Flags: Read / Write

Since: 3.0.0

The "source-function" property

```
"source-function"          char~*
```

The source function.

Owner: AgsFunction

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Chapter 7

AgsSolverMatrix

AgsSolverMatrix — solver matrix

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SOLVER_MATRIX_GET_OBJ_MUTEX()
gchar *	ags_solver_matrix_get_source_function ()
void	ags_solver_matrix_set_source_function ()
guint	ags_solver_matrix_get_row_count ()
guint	ags_solver_matrix_get_column_count ()
void	ags_solver_matrix_parse ()
void	ags_solver_matrix_insert_vector ()
void	ags_solver_matrix_remove_vector ()
void	ags_solver_matrix_eliminate ()
AgsSolverMatrix *	ags_solver_matrix_new ()
#define	AGS_IS_SOLVER_MATRIX()
#define	AGS_IS_SOLVER_MATRIX_CLASS()
#define	AGS_SOLVER_MATRIX()
#define	AGS_SOLVER_MATRIX_CLASS()
#define	AGS_SOLVER_MATRIX_GET_CLASS()
GType	ags_solver_matrix_get_type ()

Properties

guint	column-count	Read
guint	row-count	Read
char *	source-function	Read / Write

Types and Values

#define	AGS_TYPE_SOLVER_MATRIX
struct	AgsSolverMatrix
struct	AgsSolverMatrixClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsSolverMatrix
```

Includes

```
#include <ags/lib/ags_solver_matrix.h>
```

Description

The AgsSolverMatrix contains the solver vectors.

Functions

AGS_SOLVER_MATRIX_GET_OBJ_MUTEX()

```
#define AGS_SOLVER_MATRIX_GET_OBJ_MUTEX(obj) (&(((AgsSolverMatrix *) obj)->obj_mutex))
```

ags_solver_matrix_get_source_function ()

```
gchar~*
ags_solver_matrix_get_source_function (AgsSolverMatrix *solver_matrix);
```

Get column count of *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix

Returns

the column count

Since: 3.9.3

ags_solver_matrix_set_source_function ()

```
void
ags_solver_matrix_set_source_function (AgsSolverMatrix *solver_matrix,
                                       gchar *source_function);
```

Set *source_function* of *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix	
source_function	the source function	

Since: 3.9.3

ags_solver_matrix_get_row_count ()

```
guint
ags_solver_matrix_get_row_count (AgsSolverMatrix *solver_matrix);
```

Get row count of *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix	

Returns

the row count

Since: 3.9.3

ags_solver_matrix_get_column_count ()

```
guint
ags_solver_matrix_get_column_count (AgsSolverMatrix *solver_matrix);
```

ags_solver_matrix_parse ()

```
void
ags_solver_matrix_parse (AgsSolverMatrix *solver_matrix,
                         gchar *source_function);
```

Parse *source_function* and apply to *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix	
source_function	the normalized source function	

Since: 3.9.3

ags_solver_matrix_insert_vector ()

```
void
ags_solver_matrix_insert_vector (AgsSolverMatrix *solver_matrix,
                                 AgsSolverVector *solver_vector,
                                 gint position);
```

Insert *solver_vector* to *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix	
solver_vector	the AgsSolverVector	
position	the position	

Since: 3.2.0

ags_solver_matrix_remove_vector ()

```
void
ags_solver_matrix_remove_vector (AgsSolverMatrix *solver_matrix,
                                 AgsSolverVector *solver_vector);
```

Remove *solver_vector* from *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix	
solver_vector	the AgsSolverVector	

Since: 3.2.0

ags_solver_matrix_eliminate ()

```
void
ags_solver_matrix_eliminate (AgsSolverMatrix *solver_matrix,
                             guint nth_column,
                             guint nth_row_a,
                             guint nth_row_b);
```

Eliminate one column of *solver_matrix* .

Parameters

solver_matrix	the AgsSolverMatrix	
nth_column	the nth column	
nth_row_a	the nth row as a	
nth_row_b	the nth row as b	

Since: 3.9.3

ags_solver_matrix_new ()

```
AgsSolverMatrix~*
ags_solver_matrix_new ();
```

Instantiate a new AgsSolverMatrix.

Returns

the new instance

Since: 3.0.0

AGS_IS_SOLVER_MATRIX()

```
#define AGS_IS_SOLVER_MATRIX(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SOLVER_MATRIX))
```

AGS_IS_SOLVER_MATRIX_CLASS()

```
#define AGS_IS_SOLVER_MATRIX_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SOLVER_MATRIX))
```

AGS_SOLVER_MATRIX()

```
#define AGS_SOLVER_MATRIX(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOLVER_MATRIX, AgsSolverMatrix))
```

AGS_SOLVER_MATRIX_CLASS()

```
#define AGS_SOLVER_MATRIX_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SOLVER_MATRIX, AgsSolverMatrixClass))
```

AGS_SOLVER_MATRIX_GET_CLASS()

```
#define AGS_SOLVER_MATRIX_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_SOLVER_MATRIX, AgsSolverMatrixClass))
```

ags_solver_matrix_get_type ()

```
GType
ags_solver_matrix_get_type (void);
```

Types and Values

AGS_TYPE_SOLVER_MATRIX

```
#define AGS_TYPE_SOLVER_MATRIX                 (ags_solver_matrix_get_type())
```

struct AgsSolverMatrix

```
struct AgsSolverMatrix;
```

struct AgsSolverMatrixClass

```
struct AgsSolverMatrixClass {
  GObjectClass gobject;
};
```

Property Details

The "column-count" property

```
  "column-count"              guint
```

The column count of the term table.

Owner: AgsSolverMatrix

Flags: Read

Default value: 0

Since: 3.9.3

The "row-count" property

```
  "row-count"                 guint
```

The row count of the term table.

Owner: AgsSolverMatrix

Flags: Read

Default value: 0

Since: 3.9.3

The "source-function" property

```
  "source-function"           char~*
```

The assigned source-function.

Owner: AgsSolverMatrix

Flags: Read / Write

Default value: NULL

Since: 3.9.3

Chapter 8

AgsSolverVector

AgsSolverVector — solver vector

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SOLVER_VECTOR_GET_OBJ_MUTEX()
gchar *	ags_solver_vector_get_source_polynomial ()
void	ags_solver_vector_set_source_polynomial ()
guint	ags_solver_vector_get_polynomial_count ()
void	ags_solver_vector_insert_polynomial ()
void	ags_solver_vector_remove_polynomial ()
AgsSolverVector *	ags_solver_vector_new ()
#define	AGS_IS_SOLVER_VECTOR()
#define	AGS_IS_SOLVER_VECTOR_CLASS()
#define	AGS_SOLVER_VECTOR()
#define	AGS_SOLVER_VECTOR_CLASS()
#define	AGS_SOLVER_VECTOR_GET_CLASS()
GType	ags_solver_vector_get_type ()

Properties

guint	polynomial-count	Read
char *	source-polynomial	Read / Write

Types and Values

#define	AGS_TYPE_SOLVER_VECTOR
struct	AgsSolverVector
struct	AgsSolverVectorClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsSolverVector
```

Includes

```
#include <ags/lib/ags_solver_vector.h>
```

Description

The AgsSolverVector contains the solver vectors.

Functions

AGS_SOLVER_VECTOR_GET_OBJ_MUTEX()

```
#define AGS_SOLVER_VECTOR_GET_OBJ_MUTEX(obj) (&(((AgsSolverVector *) obj)->obj_mutex))
```

ags_solver_vector_get_source_polynomial ()

```
gchar~*
ags_solver_vector_get_source_polynomial
                                (AgsSolverVector *solver_vector);
```

Get source polynomial of *solver_vector* .

Parameters

solver_vector	the AgsSolverVector

Returns

the source polynomial

Since: 3.9.3

ags_solver_vector_set_source_polynomial ()

```
void
ags_solver_vector_set_source_polynomial
                                (AgsSolverVector *solver_vector,
                                 gchar *source_polynomial);
```

Set *source_polynomial* of *solver_vector* .

Parameters

solver_vector	the AgsSolverVector	
source_polynomial	the source polynomial	

Since: 3.9.3

ags_solver_vector_get_polynomial_count ()

```
guint
ags_solver_vector_get_polynomial_count
                                (AgsSolverVector *solver_vector);
```

Get polynomial count of *solver_vector* .

Parameters

solver_vector	the AgsSolverVector	

Returns

the polynomial count

Since: 3.9.3

ags_solver_vector_insert_polynomial ()

```
void
ags_solver_vector_insert_polynomial (AgsSolverVector *solver_vector,
                                     AgsSolverPolynomial *solver_polynomial,
                                     gint position);
```

Insert *solver_polynomial* to *solver_vector* .

Parameters

solver_vector	the AgsSolverVector	
solver_polynomial	the AgsSolverPolynomial	
position	the position	

Since: 3.2.0

ags_solver_vector_remove_polynomial ()

```
void
ags_solver_vector_remove_polynomial (AgsSolverVector *solver_vector,
                                     AgsSolverPolynomial *solver_polynomial);
```

Remove *solver_polynomial* from *solver_vector* .

Parameters

solver_vector	the AgsSolverVector	
solver_polynomial	the AgsSolverPolynomial	

Since: 3.2.0

ags_solver_vector_new ()

```
AgsSolverVector~*
ags_solver_vector_new ();
```

Instantiate a new AgsSolverVector.

Returns

the new instance

Since: 3.0.0

AGS_IS_SOLVER_VECTOR()

```
#define AGS_IS_SOLVER_VECTOR(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SOLVER_VECTOR))
```

AGS_IS_SOLVER_VECTOR_CLASS()

```
#define AGS_IS_SOLVER_VECTOR_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SOLVER_VECTOR))
```

AGS_SOLVER_VECTOR()

```
#define AGS_SOLVER_VECTOR(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOLVER_VECTOR, AgsSolverVector))
```

AGS_SOLVER_VECTOR_CLASS()

```
#define AGS_SOLVER_VECTOR_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SOLVER_VECTOR, AgsSolverVectorClass))
```

AGS_SOLVER_VECTOR_GET_CLASS()

```
#define AGS_SOLVER_VECTOR_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_SOLVER_VECTOR, AgsSolverVectorClass))
```

ags_solver_vector_get_type ()

```
GType
ags_solver_vector_get_type (void);
```

Types and Values

AGS_TYPE_SOLVER_VECTOR

```
#define AGS_TYPE_SOLVER_VECTOR                (ags_solver_vector_get_type())
```

struct AgsSolverVector

```
struct AgsSolverVector;
```

struct AgsSolverVectorClass

```
struct AgsSolverVectorClass {
  GObjectClass gobject;
};
```

Property Details

The "polynomial-count" property

```
  "polynomial-count"         guint
```

The polynomial count of the term table.

Owner: AgsSolverVector

Flags: Read

Default value: 0

Since: 3.9.3

The "source-polynomial" property

```
  "source-polynomial"        char~*
```

The assigned source-polynomial.

Owner: AgsSolverVector

Flags: Read / Write

Default value: NULL

Since: 3.9.3

Chapter 9

AgsSolverPolynomial

AgsSolverPolynomial — solver polynomial

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SOLVER_POLYNOMIAL_GET_OBJ_MUTEX()
GQuark	ags_solver_polynomial_error_quark ()
void	ags_solver_polynomial_set_polynomial ()
gchar *	ags_solver_polynomial_get_polynomial ()
void	ags_solver_polynomial_set_coefficient ()
gchar *	ags_solver_polynomial_get_coefficient ()
gchar **	ags_solver_polynomial_get_symbol ()
gchar **	ags_solver_polynomial_get_exponent ()
void	ags_solver_polynomial_set_coefficient_value ()
AgsComplex *	ags_solver_polynomial_get_coefficient_value ()
AgsComplex *	ags_solver_polynomial_get_exponent_value ()
void	ags_solver_polynomial_update ()
void	ags_solver_polynomial_parse ()
AgsSolverPolynomial *	ags_solver_polynomial_add ()
AgsSolverPolynomial *	ags_solver_polynomial_subtract ()
AgsSolverPolynomial *	ags_solver_polynomial_multiply ()
AgsSolverPolynomial *	ags_solver_polynomial_divide ()
AgsSolverPolynomial *	ags_solver_polynomial_raise_power ()
AgsSolverPolynomial *	ags_solver_polynomial_extract_root ()
AgsSolverPolynomial *	ags_solver_polynomial_new ()
#define	AGS_IS_SOLVER_POLYNOMIAL()
#define	AGS_IS_SOLVER_POLYNOMIAL_CLASS()
#define	AGS_SOLVER_POLYNOMIAL()
#define	AGS_SOLVER_POLYNOMIAL_CLASS()
#define	AGS_SOLVER_POLYNOMIAL_GET_CLASS()
GType	ags_solver_polynomial_get_type ()

Properties

char *	coefficient	Read / Write
AgsComplex *	coefficient-value	Read / Write
gpointer	exponent	Read
gpointer	exponent-value	Read
char *	polynomial	Read / Write
gpointer	symbol	Read / Write

Types and Values

#define	AGS_SOLVER_POLYNOMIAL_ERROR
enum	AgsSolverPolynomialError
#define	AGS_TYPE_SOLVER_POLYNOMIAL
struct	AgsSolverPolynomial
struct	AgsSolverPolynomialClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsSolverPolynomial
```

Includes

```
#include <ags/lib/ags_solver_polynomial.h>
```

Description

The AgsSolverPolynomial contains the solver polynomials.

Functions

AGS_SOLVER_POLYNOMIAL_GET_OBJ_MUTEX()

```
#define AGS_SOLVER_POLYNOMIAL_GET_OBJ_MUTEX(obj) (&(((AgsSolverPolynomial *) obj)-> ↩
    obj_mutex))
```

ags_solver_polynomial_error_quark ()

```
GQuark
ags_solver_polynomial_error_quark ();
```

ags_solver_polynomial_set_polynomial ()

```
void
ags_solver_polynomial_set_polynomial (AgsSolverPolynomial *solver_polynomial,
                                      gchar *polynomial);
```

Set *polynomial* of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	
polynomial	the polynomial	

Since: 3.9.3

ags_solver_polynomial_get_polynomial ()

```
gchar~*
ags_solver_polynomial_get_polynomial (AgsSolverPolynomial *solver_polynomial);
```

Get polynomial of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	

Returns

the polynomial

Since: 3.9.3

ags_solver_polynomial_set_coefficient ()

```
void
ags_solver_polynomial_set_coefficient (AgsSolverPolynomial *solver_polynomial,
                                       gchar *coefficient);
```

Set *coefficient* of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	
coefficient	the coefficient	

Since: 3.9.3

ags_solver_polynomial_get_coefficient ()

```
gchar~*
ags_solver_polynomial_get_coefficient (AgsSolverPolynomial *solver_polynomial);
```

Get coefficient of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial

Returns

the coefficient

Since: 3.9.3

ags_solver_polynomial_get_symbol ()

```
gchar~**
ags_solver_polynomial_get_symbol (AgsSolverPolynomial *solver_polynomial);
```

Get symbol of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial

Returns

the symbol.

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

Since: 3.9.3

ags_solver_polynomial_get_exponent ()

```
gchar~**
ags_solver_polynomial_get_exponent (AgsSolverPolynomial *solver_polynomial);
```

Get exponent of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial

Returns

the exponent.

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

Since: 3.9.3

ags_solver_polynomial_set_coefficient_value ()

```
void
ags_solver_polynomial_set_coefficient_value
                                (AgsSolverPolynomial *solver_polynomial,
                                 AgsComplex *coefficient_value);
```

Set *coefficient_value* of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	
coefficient_value	the coefficient value	

Since: 3.9.3

ags_solver_polynomial_get_coefficient_value ()

```
AgsComplex~*
ags_solver_polynomial_get_coefficient_value
                                (AgsSolverPolynomial *solver_polynomial);
```

Get coefficient value of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	

Returns

the coefficient value

Since: 3.9.3

ags_solver_polynomial_get_exponent_value ()

```
AgsComplex~*
ags_solver_polynomial_get_exponent_value
                                (AgsSolverPolynomial *solver_polynomial);
```

Get exponent value of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	

Returns

the exponent value.

[element-type gpointer][array zero-terminated=0][transfer none]

Since: 3.9.3

ags_solver_polynomial_update ()

```
void
ags_solver_polynomial_update (AgsSolverPolynomial *solver_polynomial);
```

Update string representation of *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	

Since: 3.2.0

ags_solver_polynomial_parse ()

```
void
ags_solver_polynomial_parse (AgsSolverPolynomial *solver_polynomial,
                             gchar *polynomial);
```

Parse *polynomial* and apply to *solver_polynomial* .

Parameters

solver_polynomial	the AgsSolverPolynomial	
polynomial	the polynomial as string	

Since: 3.2.0

ags_solver_polynomial_add ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_add (AgsSolverPolynomial *polynomial_a,
                           AgsSolverPolynomial *polynomial_b,
                           GError **error);
```

Perform addition of *polynomial_a* and *polynomial_b* . Both summands need to have the very same symbol and exponent, otherwise NULL returned and error is appropriately set.

Parameters

polynomial_a	the first summand	
polynomial_b	the second summand	
error	return location of GError	

Returns

the newly instantiated AgsSolverPolynomial or NULL.

[transfer full]

Since: 3.2.0

ags_solver_polynomial_subtract ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_subtract (AgsSolverPolynomial *polynomial_a,
                                AgsSolverPolynomial *polynomial_b,
                                GError **error);
```

Perform subtraction of *polynomial_a* and *polynomial_b* . Both minuend and subtrahend need to have the very same symbol and exponent, otherwise NULL returned and error is appropriately set.

Parameters

polynomial_a	the minuend	
polynomial_b	the subtrahend	
error	return location of GError	

Returns

the newly instantiated AgsSolverPolynomial or NULL.

[transfer full]

Since: 3.2.0

ags_solver_polynomial_multiply ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_multiply (AgsSolverPolynomial *polynomial_a,
                                AgsSolverPolynomial *polynomial_b,
                                GError **error);
```

Perform multiplication of *polynomial_a* and *polynomial_b* .

Parameters

polynomial_a	the first factor	
polynomial_b	the second factor	
error	return location of GError	

Returns

the newly instantiated AgsSolverPolynomial or NULL.

[transfer full]

Since: 3.2.0

ags_solver_polynomial_divide ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_divide (AgsSolverPolynomial *polynomial_a,
                              AgsSolverPolynomial *polynomial_b,
                              GError **error);
```

Perform division of *polynomial_a* and *polynomial_b* .

Parameters

polynomial_a	the dividend	
polynomial_b	the divisor	
error	return location of GError	

Returns

the newly instantiated AgsSolverPolynomial or NULL.

[transfer full]

Since: 3.2.0

ags_solver_polynomial_raise_power ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_raise_power (AgsSolverPolynomial *polynomial_a,
                                   AgsSolverPolynomial *polynomial_b,
                                   GError **error);
```

Perform raising power of *polynomial_a* and *polynomial_b* .

Parameters

polynomial_a	the base	
polynomial_b	the exponent	
error	return location of GError	

Returns

the newly instantiated AgsSolverPolynomial or NULL.

[transfer full]

Since: 3.2.0

ags_solver_polynomial_extract_root ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_extract_root (AgsSolverPolynomial *polynomial_a,
                                    AgsSolverPolynomial *polynomial_b,
                                    GError **error);
```

Perform raising power of *polynomial_a* and *polynomial_b* .

Parameters

polynomial_a	the base	
polynomial_b	the index of root	
error	return location of GError	

Returns

the newly instantiated AgsSolverPolynomial or NULL.

[transfer full]

Since: 3.2.0

ags_solver_polynomial_new ()

```
AgsSolverPolynomial~*
ags_solver_polynomial_new ();
```

Instantiate a new AgsSolverPolynomial.

Returns

the new instance

Since: 3.2.0

AGS_IS_SOLVER_POLYNOMIAL()

```
#define AGS_IS_SOLVER_POLYNOMIAL(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SOLVER_POLYNOMIAL))
```

AGS_IS_SOLVER_POLYNOMIAL_CLASS()

```
#define AGS_IS_SOLVER_POLYNOMIAL_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SOLVER_POLYNOMIAL))
```

AGS_SOLVER_POLYNOMIAL()

```
#define AGS_SOLVER_POLYNOMIAL(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOLVER_POLYNOMIAL, AgsSolverPolynomial))
```

AGS_SOLVER_POLYNOMIAL_CLASS()

```
#define AGS_SOLVER_POLYNOMIAL_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_SOLVER_POLYNOMIAL, AgsSolverPolynomialClass))
```

AGS_SOLVER_POLYNOMIAL_GET_CLASS()

```
#define AGS_SOLVER_POLYNOMIAL_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS (obj,  ↩
    AGS_TYPE_SOLVER_POLYNOMIAL, AgsSolverPolynomialClass))
```

ags_solver_polynomial_get_type ()

```
GType
ags_solver_polynomial_get_type (void);
```

Types and Values

AGS_SOLVER_POLYNOMIAL_ERROR

```
#define AGS_SOLVER_POLYNOMIAL_ERROR (ags_solver_polynomial_error_quark())
```

enum AgsSolverPolynomialError

Members

AGS_SOLVER_POLYNOMIAL_ERROR_SYMBOL_MISMATCH	
AGS_SOLVER_POLYNOMIAL_ERROR_EXPONENT_MISMATCH	
AGS_SOLVER_POLYNOMIAL_ERROR_DIVISION_BY_ZERO	
AGS_SOLVER_POLYNOMIAL_ERROR_EXPONENT_NOT_CONSTANT	

AGS_TYPE_SOLVER_POLYNOMIAL

```
#define AGS_TYPE_SOLVER_POLYNOMIAL                (ags_solver_polynomial_get_type())
```

struct AgsSolverPolynomial

```
struct AgsSolverPolynomial;
```

struct AgsSolverPolynomialClass

```
struct AgsSolverPolynomialClass {
  GObjectClass gobject;
};
```

Property Details

The "coefficient" property

```
  "coefficient"              char~*
```

The assigned coefficient.

Owner: AgsSolverPolynomial

Flags: Read / Write

Default value: NULL

Since: 3.2.0

The "coefficient-value" property

```
"coefficient-value"        AgsComplex~*
```

Coefficient value.

Owner: AgsSolverPolynomial

Flags: Read / Write

Since: 3.2.0

The "exponent" property

```
"exponent"                 gpointer
```

The assigned exponent.

Owner: AgsSolverPolynomial

Flags: Read

Since: 3.2.0

The "exponent-value" property

```
"exponent-value"           gpointer
```

Exponent value.

Owner: AgsSolverPolynomial

Flags: Read

Since: 3.2.0

The "polynomial" property

```
"polynomial"               char~*
```

The assigned polynomial.

Owner: AgsSolverPolynomial

Flags: Read / Write

Default value: NULL

Since: 3.2.0

The "symbol" property

```
"symbol"                   gpointer
```

The assigned symbol.

Owner: AgsSolverPolynomial

Flags: Read / Write

Since: 3.2.0

Chapter 10

AgsLog

AgsLog — Log

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LOG_GET_OBJ_MUTEX()
void	ags_log_add_message ()
GList *	ags_log_get_messages ()
AgsLog *	ags_log_get_instance ()
AgsLog *	ags_log_new ()
#define	AGS_IS_LOG()
#define	AGS_IS_LOG_CLASS()
#define	AGS_LOG()
#define	AGS_LOG_CLASS()
#define	AGS_LOG_GET_CLASS()
GType	ags_log_get_type ()

Types and Values

#define	AGS_TYPE_LOG
struct	AgsLog
struct	AgsLogClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsLog
```

Includes

```
#include <ags/lib/ags_log.h>
```

Description

The AgsLog logging class.

Functions

AGS_LOG_GET_OBJ_MUTEX()

```
#define AGS_LOG_GET_OBJ_MUTEX(obj) (&(((AgsLog *) obj)->obj_mutex))
```

ags_log_add_message ()

```
void
ags_log_add_message (AgsLog *log,
                     gchar *str);
```

Add a message to *log* .

Parameters

log	the AgsLog	
str	the message.	*[transfer full]*

Since: 3.0.0

ags_log_get_messages ()

```
GList~*
ags_log_get_messages (AgsLog *log);
```

Get log messages as GList containing strings.

Parameters

log	the AgsLog	

Returns

the GList containing log messages.

[element-type utf8][transfer none]

Since: 3.0.0

ags_log_get_instance ()

```
AgsLog~*
ags_log_get_instance ();
```

Get your logging instance.

Returns

the AgsLog instance.

[transfer none]

Since: 3.0.0

ags_log_new ()

```
AgsLog~*
ags_log_new ();
```

Instantiate a new AgsLog.

Returns

the new instance

Since: 3.0.0

AGS_IS_LOG()

```
#define AGS_IS_LOG(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_LOG))
```

AGS_IS_LOG_CLASS()

```
#define AGS_IS_LOG_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_LOG))
```

AGS_LOG()

```
#define AGS_LOG(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_LOG, AgsLog ↩
    ))
```

AGS_LOG_CLASS()

```
#define AGS_LOG_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_LOG,  ↩
    AgsLogClass))
```

AGS_LOG_GET_CLASS()

```
#define AGS_LOG_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS (obj, AGS_TYPE_LOG,  ↩
    AgsLogClass))
```

ags_log_get_type ()

```
GType
ags_log_get_type (void);
```

Types and Values

AGS_TYPE_LOG

```
#define AGS_TYPE_LOG                (ags_log_get_type())
```

struct AgsLog

```
struct AgsLog;
```

struct AgsLogClass

```
struct AgsLogClass {
  GObjectClass gobject;
};
```

Chapter 11

AgsRegex

AgsRegex — coding-neutral regex

Stability Level

Stable, unless otherwise indicated

Functions

int	ags_regcomp ()
int	ags_regexec ()

Includes

```
#include <ags/lib/ags_regex.h>
```

Description

Wrapper around glibc regex functions to provide language coding-neutral processing.

Functions

ags_regcomp ()

```
int
ags_regcomp (regex_t *preg,
             const char *regex,
             int cflags);
```

Compile regular expression language independent. It sets temporaly the locale to C and then reverts it.

Parameters

preg	pattern buffer storage area	
regex	regular expression as string	
cflags	flags	

Since: 3.0.0

ags_regexec ()

```
int
ags_regexec (const regex_t *preg,
             const char *string,
             size_t nmatch,
             regmatch_t pmatch[],
             int eflags);
```

Execute regular expression language independent. It sets temporaly the locale to C and then reverts it.

Parameters

preg	pattern buffer storage area	
string	the string to match	
nmatch	match count	
pmatch	match array	
eflags	error flags	

Since: 3.0.0

Chapter 12

AgsStringUtil

AgsStringUtil — string util

Stability Level

Stable, unless otherwise indicated

Functions

gchar *	ags_string_util_escape_single_quote ()
guint	ags_strv_length ()
gboolean	ags_strv_contains ()
gint	ags_strv_index ()
gboolean	ags_strv_equal ()
GType	ags_string_util_get_type ()

Types and Values

#define	AGS_TYPE_STRING_UTIL
struct	AgsStringUtil

Object Hierarchy

```
    GBoxed
    &#x2570;&#x2500;&#x2500; AgsStringUtil
```

Includes

```
#include <ags/lib/ags_string_util.h>
```

Description

Common string utility functions.

Functions

ags_string_util_escape_single_quote ()

```
gchar~*
ags_string_util_escape_single_quote (gchar *str);
```

Escape all occurence of single quotes.

Parameters

str	the string to escape	

Returns

the newly allocated string.

[transfer full]

Since: 3.0.0

ags_strv_length ()

```
guint
ags_strv_length (gchar **str_array);
```

Count the number of non-NULL entries in the array.

Parameters

str_array	the string vector.	*[element-type utf8][array zero-terminated=1][transfer none]*

Returns

the length of the vector

Since: 3.0.0

ags_strv_contains ()

```
gboolean
ags_strv_contains (gchar **str_array,
                   gchar *str);
```

Check occurence of `str` within `str_array`

Parameters

str_array	the string vector.	*[element-type utf8][array zero-terminated=1][transfer none]*
str	the string to match	

Returns

TRUE if found, else FALSE

Since: 3.0.0

ags_strv_index ()

```
gint
ags_strv_index (gchar **str_array,
                gchar *str);
```

Check by comparing *str* against *str_array* items.

Parameters

str_array	the string vector.	*[element-type utf8][array zero-terminated=1][transfer none]*
str	the string to match	

Returns

the first matching position or -1, if not found

Since: 3.0.0

ags_strv_equal ()

```
gboolean
ags_strv_equal (gchar **strv1,
                gchar **strv2);
```

Check *strv1* and *strv2* to be equal. The arrays don't have to be sorted.

Parameters

strv1	the string vector	
strv2	an other string vector	

Returns

TRUE on success, otherwise FALSE

Since: 3.2.1

ags_string_util_get_type ()

```
GType
ags_string_util_get_type (void);
```

Types and Values

AGS_TYPE_STRING_UTIL

```
#define AGS_TYPE_STRING_UTIL         (ags_string_util_get_type())
```

struct AgsStringUtil

```
struct AgsStringUtil {
  //empty
};
```

Chapter 13

AgsTime

AgsTime — utility functions of time

Stability Level

Stable, unless otherwise indicated

Functions

gchar *	ags_time_get_uptime_from_offset ()
gboolean	ags_time_timeout_expired ()

Types and Values

#define	AGS_TIME_ZERO
#define	AGS_USEC_PER_SEC
#define	AGS_NSEC_PER_SEC

Includes

```
#include <ags/lib/ags_time.h>
```

Description

Functions to help you dealing with time.

Functions

ags_time_get_uptime_from_offset ()

```
gchar~*
ags_time_get_uptime_from_offset (guint offset,
                                 gdouble bpm,
                                 gdouble delay,
                                 gdouble delay_factor);
```

Calculates uptime from *offset* by applying factors *bpm* , *delay* and *delay_factor* giving you the result as string.

Parameters

offset	the offset in tics	
bpm	the beats per minute	
delay	the buffer time as delay	
delay_factor	tact segmentation	

Returns

the string containing the time, a minus sign is prepend if factors are invalid.

[transfer full]

Since: 3.0.0

ags_time_timeout_expired ()

```
gboolean
ags_time_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

Types and Values

AGS_TIME_ZERO

```
#define AGS_TIME_ZERO "0000:00.000"
```

AGS_USEC_PER_SEC

```
#define AGS_USEC_PER_SEC    (1000000) /* The number of msecs per sec. */
```

AGS_NSEC_PER_SEC

```
#define AGS_NSEC_PER_SEC    (1000000000) /* The number of nsecs per sec. */
```

Chapter 14

AgsTurtle

AgsTurtle — Terse RDF Triple Language

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TURTLE_GET_OBJ_MUTEX()
gchar *	ags_turtle_read_iriref ()
gchar *	ags_turtle_read_pname_ns ()
gchar *	ags_turtle_read_pname_ln ()
gchar *	ags_turtle_read_blank_node_label ()
gchar *	ags_turtle_read_langtag ()
gboolean	ags_turtle_match_iriref ()
gboolean	ags_turtle_match_pname_ns ()
gboolean	ags_turtle_match_pname_ln ()
gboolean	ags_turtle_match_blank_node_label ()
gboolean	ags_turtle_match_langtag ()
gchar *	ags_turtle_read_boolean ()
gchar *	ags_turtle_read_integer ()
gchar *	ags_turtle_read_decimal ()
gchar *	ags_turtle_read_double ()
gchar *	ags_turtle_read_exponent ()
gboolean	ags_turtle_match_boolean ()
gboolean	ags_turtle_match_integer ()
gboolean	ags_turtle_match_decimal ()
gboolean	ags_turtle_match_double ()
gboolean	ags_turtle_match_exponent ()
gchar *	ags_turtle_read_string ()
gboolean	ags_turtle_match_string ()
gchar *	ags_turtle_read_string_literal_quote ()
gchar *	ags_turtle_read_string_literal_single_quote ()
gchar *	ags_turtle_read_string_literal_long_quote ()
gchar *	ags_turtle_read_string_literal_long_single_quote ()
gboolean	ags_turtle_match_string_literal_quote ()

gboolean	ags_turtle_match_string_literal_single_quote ()
gboolean	ags_turtle_match_string_literal_long_quote ()
gboolean	ags_turtle_match_string_literal_long_single_quote ()
gchar *	ags_turtle_read_uchar ()
gchar *	ags_turtle_read_echar ()
gchar *	ags_turtle_read_ws ()
gchar *	ags_turtle_read_anon ()
gchar *	ags_turtle_read_pn_chars_base ()
gchar *	ags_turtle_read_pn_chars_u ()
gchar *	ags_turtle_read_pn_chars ()
gchar *	ags_turtle_read_pn_prefix ()
gchar *	ags_turtle_read_pn_local ()
gchar *	ags_turtle_read_plx ()
gchar *	ags_turtle_read_percent ()
gchar *	ags_turtle_read_hex ()
gchar *	ags_turtle_read_pn_local_esc ()
gboolean	ags_turtle_match_uchar ()
gboolean	ags_turtle_match_echar ()
gboolean	ags_turtle_match_ws ()
gboolean	ags_turtle_match_anon ()
gboolean	ags_turtle_match_pn_chars_base ()
gboolean	ags_turtle_match_pn_chars_u ()
gboolean	ags_turtle_match_pn_chars ()
gboolean	ags_turtle_match_pn_prefix ()
gboolean	ags_turtle_match_pn_local ()
gboolean	ags_turtle_match_plx ()
gboolean	ags_turtle_match_percent ()
gboolean	ags_turtle_match_hex ()
gboolean	ags_turtle_match_pn_local_esc ()
GList *	ags_turtle_find_xpath ()
GList *	ags_turtle_find_xpath_with_context_node ()
gchar *	ags_turtle_string_convert ()
xmlDoc *	ags_turtle_load ()
AgsTurtle *	ags_turtle_new ()
#define	AGS_IS_TURTLE()
#define	AGS_IS_TURTLE_CLASS()
#define	AGS_TURTLE()
#define	AGS_TURTLE_CLASS()
#define	AGS_TURTLE_GET_CLASS()
GType	ags_turtle_flags_get_type ()
GType	ags_turtle_get_type ()

Properties

char *	filename	Read / Write
gpointer	xml-doc	Read / Write

Types and Values

#define	AGS_TURTLE_DEFAULT_ENCODING
#define	AGS_TURTLE_DEFAULT_VERSION
#define	AGS_TURTLE_BOOLEAN_LITERAL_TRUE

#define	AGS_TURTLE_BOOLEAN_LITERAL_FALSE
enum	AgsTurtleFlags
#define	AGS_TYPE_TURTLE
#define	AGS_TYPE_TURTLE_FLAGS
struct	AgsTurtle
struct	AgsTurtleClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsTurtleFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsTurtle
```

Includes

```
#include <ags/lib/ags_turtle.h>
```

Description

The AgsTurtle object converts Terse RDF Triple Language files into XML.

Functions

AGS_TURTLE_GET_OBJ_MUTEX()

```
#define AGS_TURTLE_GET_OBJ_MUTEX(obj) (&(((AgsTurtle *) obj)->obj_mutex))
```

ags_turtle_read_iriref ()

```
gchar~*
ags_turtle_read_iriref (gchar *offset,
                        gchar *end_ptr);
```

Read iriref value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pname_ns ()

```
gchar~*
ags_turtle_read_pname_ns (gchar *offset,
                          gchar *end_ptr);
```

Read prefixed-name namespace value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pname_ln ()

```
gchar~*
ags_turtle_read_pname_ln (gchar *offset,
                          gchar *end_ptr);
```

Read prefixed-name localized name value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_blank_node_label ()

```
gchar~*
ags_turtle_read_blank_node_label (gchar *offset,
                                  gchar *end_ptr);
```

Read blank node label value.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_langtag ()

```
gchar~*
ags_turtle_read_langtag (gchar *offset,
                         gchar *end_ptr);
```

Read langtag value.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_match_iriref ()

```
gboolean
ags_turtle_match_iriref (gchar *offset,
                         gchar *end_ptr,
                         gchar **start_offset,
                         gchar **end_offset);
```

Match iriref.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pname_ns ()

```
gboolean
ags_turtle_match_pname_ns (gchar *offset,
                           gchar *end_ptr,
                           gchar **start_offset,
                           gchar **end_offset);
```

Match match pname-ns.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pname_ln ()

```
gboolean
ags_turtle_match_pname_ln (gchar *offset,
                           gchar *end_ptr,
                           gchar **start_offset,
                           gchar **end_offset);
```

Match pname-ln.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_blank_node_label ()

```
gboolean
ags_turtle_match_blank_node_label (gchar *offset,
                                   gchar *end_ptr,
                                   gchar **start_offset,
                                   gchar **end_offset);
```

Match blank node label.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_langtag ()

```
gboolean
ags_turtle_match_langtag (gchar *offset,
                          gchar *end_ptr,
                          gchar **start_offset,
                          gchar **end_offset);
```

Match langtag.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_read_boolean ()

```
gchar~*
ags_turtle_read_boolean (gchar *offset,
                         gchar *end_ptr);
```

Read boolean value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_integer ()

```
gchar~*
ags_turtle_read_integer (gchar *offset,
                         gchar *end_ptr);
```

Read integer value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_decimal ()

```
gchar~*
ags_turtle_read_decimal (gchar *offset,
                         gchar *end_ptr);
```

Read decimal value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_double ()

```
gchar~*
ags_turtle_read_double (gchar *offset,
                        gchar *end_ptr);
```

Read double value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_exponent ()

```
gchar~*
ags_turtle_read_exponent (gchar *offset,
                          gchar *end_ptr);
```

Read exponent value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_match_boolean ()

```
gboolean
ags_turtle_match_boolean (gchar *offset,
                          gchar *end_ptr,
                          gchar **start_offset,
                          gchar **end_offset);
```

Match boolean.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_integer ()

```
gboolean
ags_turtle_match_integer (gchar *offset,
                          gchar *end_ptr,
                          gchar **start_offset,
                          gchar **end_offset);
```

Match integer.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_decimal ()

```
gboolean
ags_turtle_match_decimal (gchar *offset,
                          gchar *end_ptr,
                          gchar **start_offset,
                          gchar **end_offset);
```

Match decimal.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_double ()

```
gboolean
ags_turtle_match_double (gchar *offset,
                         gchar *end_ptr,
                         gchar **start_offset,
                         gchar **end_offset);
```

Match double.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_exponent ()

```
gboolean
ags_turtle_match_exponent (gchar *offset,
                           gchar *end_ptr,
                           gchar **start_offset,
                           gchar **end_offset);
```

Match exponent.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_read_string ()

```
gchar~*
ags_turtle_read_string (gchar *offset,
                        gchar *end_ptr);
```

Read string value.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_match_string ()

```
gboolean
ags_turtle_match_string (gchar *offset,
                         gchar *end_ptr,
                         gchar **start_offset,
                         gchar **end_offset);
```

Match string.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_read_string_literal_quote ()

```
gchar~*
ags_turtle_read_string_literal_quote (gchar *offset,
                                      gchar *end_ptr);
```

Read string literal quote value.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_string_literal_single_quote ()

```
gchar~*
ags_turtle_read_string_literal_single_quote
                                (gchar *offset,
                                 gchar *end_ptr);
```

Read string literal single quote value.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_string_literal_long_quote ()

```
gchar~*
ags_turtle_read_string_literal_long_quote
                                (gchar *offset,
                                 gchar *end_ptr);
```

Read string literal long quote value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_string_literal_long_single_quote ()

```
gchar~*
ags_turtle_read_string_literal_long_single_quote
                                (gchar *offset,
                                 gchar *end_ptr);
```

Read string literal long single quote value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_match_string_literal_quote ()

```
gboolean
ags_turtle_match_string_literal_quote (gchar *offset,
                                       gchar *end_ptr,
                                       gchar **start_offset,
                                       gchar **end_offset);
```

Match string literal quote.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_string_literal_single_quote ()

```
gboolean
ags_turtle_match_string_literal_single_quote
                                (gchar *offset,
                                 gchar *end_ptr,
                                 gchar **start_offset,
                                 gchar **end_offset);
```

Match string literal single quote.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_string_literal_long_quote ()

```
gboolean
ags_turtle_match_string_literal_long_quote
                                (gchar *offset,
                                 gchar *end_ptr,
                                 gchar **start_offset,
                                 gchar **end_offset);
```

Match string literal long quote.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_string_literal_long_single_quote ()

```
gboolean
ags_turtle_match_string_literal_long_single_quote
                                (gchar *offset,
                                 gchar *end_ptr,
                                 gchar **start_offset,
                                 gchar **end_offset);
```

Match string literal long single quote.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_read_uchar ()

```
gchar~*
ags_turtle_read_uchar (gchar *offset,
                       gchar *end_ptr);
```

Read uchar value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_echar ()

```
gchar~*
ags_turtle_read_echar (gchar *offset,
                       gchar *end_ptr);
```

Read echar value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_ws ()

```
gchar~*
ags_turtle_read_ws (gchar *offset,
                    gchar *end_ptr);
```

Read ws value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_anon ()

```
gchar~*
ags_turtle_read_anon (gchar *offset,
                      gchar *end_ptr);
```

Read anon value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pn_chars_base ()

```
gchar~*
ags_turtle_read_pn_chars_base (gchar *offset,
                               gchar *end_ptr);
```

Read prefixed-name chars base value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pn_chars_u ()

```
gchar~*
ags_turtle_read_pn_chars_u (gchar *offset,
                            gchar *end_ptr);
```

Read prefixed-name chars underscore value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pn_chars ()

```
gchar~*
ags_turtle_read_pn_chars (gchar *offset,
                          gchar *end_ptr);
```

Read prefixed-name chars value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pn_prefix ()

```
gchar~*
ags_turtle_read_pn_prefix (gchar *offset,
                           gchar *end_ptr);
```

Read prefixe-name prefix value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pn_local ()

```
gchar~*
ags_turtle_read_pn_local (gchar *offset,
                          gchar *end_ptr);
```

Read prefixed-name local value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_plx ()

```
gchar~*
ags_turtle_read_plx (gchar *offset,
                     gchar *end_ptr);
```

Read plx value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_percent ()

```
gchar~*
ags_turtle_read_percent (gchar *offset,
                         gchar *end_ptr);
```

Read percent value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_hex ()

```
gchar~*
ags_turtle_read_hex (gchar *offset,
                     gchar *end_ptr);
```

Read hex value.

Parameters

offset	the string pointer	
end_ptr	the end of *offset*	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_read_pn_local_esc ()

```
gchar~*
ags_turtle_read_pn_local_esc (gchar *offset,
                              gchar *end_ptr);
```

Read prefixed name local escapes.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	

Returns

a string on success otherwise NULL.

[transfer full]

Since: 3.0.0

ags_turtle_match_uchar ()

```
gboolean
ags_turtle_match_uchar (gchar *offset,
                        gchar *end_ptr,
                        gchar **start_offset,
                        gchar **end_offset);
```

Match uchar.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_echar ()

```
gboolean
ags_turtle_match_echar (gchar *offset,
                        gchar *end_ptr,
                        gchar **start_offset,
                        gchar **end_offset);
```

Match echar.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_ws ()

```
gboolean
ags_turtle_match_ws (gchar *offset,
                     gchar *end_ptr,
                     gchar **start_offset,
                     gchar **end_offset);
```

Match ws.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_anon ()

```
gboolean
ags_turtle_match_anon (gchar *offset,
                       gchar *end_ptr,
                       gchar **start_offset,
                       gchar **end_offset);
```

Match anon.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pn_chars_base ()

```
gboolean
ags_turtle_match_pn_chars_base (gchar *offset,
                                gchar *end_ptr,
                                gchar **start_offset,
                                gchar **end_offset);
```

Match pn chars base.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pn_chars_u ()

```
gboolean
ags_turtle_match_pn_chars_u (gchar *offset,
                             gchar *end_ptr,
                             gchar **start_offset,
                             gchar **end_offset);
```

Match pn chars u.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pn_chars ()

```
gboolean
ags_turtle_match_pn_chars (gchar *offset,
                           gchar *end_ptr,
                           gchar **start_offset,
                           gchar **end_offset);
```

Match pn chars.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pn_prefix ()

```
gboolean
ags_turtle_match_pn_prefix (gchar *offset,
                            gchar *end_ptr,
                            gchar **start_offset,
                            gchar **end_offset);
```

Match pn prefix.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pn_local ()

```
gboolean
ags_turtle_match_pn_local (gchar *offset,
                           gchar *end_ptr,
                           gchar **start_offset,
                           gchar **end_offset);
```

Match uchar.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_plx ()

```
gboolean
ags_turtle_match_plx (gchar *offset,
                      gchar *end_ptr,
                      gchar **start_offset,
                      gchar **end_offset);
```

Match plx.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_percent ()

```
gboolean
ags_turtle_match_percent (gchar *offset,
                          gchar *end_ptr,
                          gchar **start_offset,
                          gchar **end_offset);
```

Match percent.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_hex ()

```
gboolean
ags_turtle_match_hex (gchar *offset,
                      gchar *end_ptr,
                      gchar **start_offset,
                      gchar **end_offset);
```

Match hex.

Parameters

offset	the string pointer	
end_ptr	the end of `offset`	
start_offset	points to start offset of matched, otherwise NULL.	*[out][transfer none]*
end_offset	points to end offset of matched, otherwise NULL.	*[out][transfer none]*

Returns

TRUE on success, otherwise FALSE

Since: 3.2.12

ags_turtle_match_pn_local_esc ()

```
gboolean
ags_turtle_match_pn_local_esc (gchar *offset,
                               gchar *end_ptr,
                               gchar **start_offset,
                               gchar **end_offset);
```

ags_turtle_find_xpath ()

```
GList~*
ags_turtle_find_xpath (AgsTurtle *turtle,
                       gchar *xpath);
```

Lookup XPath expression withing `turtle`.

Parameters

turtle	the AgsTurtle	
xpath	a XPath expression as string	

Returns

a GList containing xmlNode.

[element-type libxml2.Node][transfer container]

Since: 3.0.0

ags_turtle_find_xpath_with_context_node ()

```
GList~*
ags_turtle_find_xpath_with_context_node
                               (AgsTurtle *turtle,
                                gchar *xpath,
                                xmlNode *context_node);
```

Lookup XPath expression from `context_node` withing `turtle`.

Parameters

turtle	the AgsTurtle	
xpath	a XPath expression as string	
context_node	a xmlNode	

Returns

a GList containing xmlNode.

[element-type libxml2.Node][transfer container]

Since: 3.0.0

ags_turtle_string_convert ()

```
gchar~*
ags_turtle_string_convert (AgsTurtle *turtle,
                           gchar *str);
```

ags_turtle_load ()

```
xmlDoc~*
ags_turtle_load (AgsTurtle *turtle,
                 GError **error);
```

Loads a RDF triple file into an XML Document.

Parameters

turtle	the AgsTurtle	
error	a pointer to a GError	

Returns

a xmlDoc pointer.

[transfer none]

Since: 3.0.0

ags_turtle_new ()

```
AgsTurtle~*
ags_turtle_new (gchar *filename);
```

Instantiate a new AgsTurtle.

Parameters

filename	the filename as string.	*[nullable]*

Returns

the new instance

Since: 3.0.0

AGS_IS_TURTLE()

```
#define AGS_IS_TURTLE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_TURTLE) ↩
    )
```

AGS_IS_TURTLE_CLASS()

```
#define AGS_IS_TURTLE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_TURTLE))
```

AGS_TURTLE()

```
#define AGS_TURTLE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_TURTLE,  ↩
    AgsTurtle))
```

AGS_TURTLE_CLASS()

```
#define AGS_TURTLE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_TURTLE,  ↩
    AgsTurtleClass))
```

AGS_TURTLE_GET_CLASS()

```
#define AGS_TURTLE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj), AGS_TYPE_TURTLE,  ↩
    AgsTurtleClass))
```

ags_turtle_flags_get_type ()

```
GType
ags_turtle_flags_get_type ();
```

ags_turtle_get_type ()

```
GType
ags_turtle_get_type (void);
```

Types and Values

AGS_TURTLE_DEFAULT_ENCODING

```
#define AGS_TURTLE_DEFAULT_ENCODING "UTF-8"
```

AGS_TURTLE_DEFAULT_VERSION

```
#define AGS_TURTLE_DEFAULT_VERSION "0.7.3"
```

AGS_TURTLE_BOOLEAN_LITERAL_TRUE

```
#define AGS_TURTLE_BOOLEAN_LITERAL_TRUE "true"
```

AGS_TURTLE_BOOLEAN_LITERAL_FALSE

```
#define AGS_TURTLE_BOOLEAN_LITERAL_FALSE "false"
```

enum AgsTurtleFlags

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

Members

AGS_TURTLE_TOLOWER	make all tags and attributes lowercase

AGS_TYPE_TURTLE

```
#define AGS_TYPE_TURTLE                 (ags_turtle_get_type())
```

AGS_TYPE_TURTLE_FLAGS

```
#define AGS_TYPE_TURTLE_FLAGS           (ags_turtle_flags_get_type())
```

struct AgsTurtle

```
struct AgsTurtle;
```

struct AgsTurtleClass

```
struct AgsTurtleClass {
  GObjectClass gobject;
};
```

Property Details

The "filename" property

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

The assigned filename.

Owner: AgsTurtle

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "xml-doc" property

```
  "xml-doc"                   gpointer
```

The assigned xml-doc.

Owner: AgsTurtle

Flags: Read / Write

Since: 3.0.0

Chapter 15

AgsTurtleManager

AgsTurtleManager — Singleton pattern to organize turtles

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TURTLE_MANAGER_GET_OBJ_MUTEX()
GObject *	ags_turtle_manager_find ()
void	ags_turtle_manager_add ()
AgsTurtleManager *	ags_turtle_manager_get_instance ()
AgsTurtleManager *	ags_turtle_manager_new ()
#define	AGS_IS_TURTLE_MANAGER()
#define	AGS_IS_TURTLE_MANAGER_CLASS()
#define	AGS_TURTLE_MANAGER()
#define	AGS_TURTLE_MANAGER_CLASS()
#define	AGS_TURTLE_MANAGER_GET_CLASS()
GType	ags_turtle_manager_get_type ()

Types and Values

#define	AGS_TYPE_TURTLE_MANAGER
struct	AgsTurtleManager
struct	AgsTurtleManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsTurtleManager
```

Includes

```
#include <ags/lib/ags_turtle_manager.h>
```

Description

The AgsTurtleManager keeps track of your turtles.

Functions

AGS_TURTLE_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_TURTLE_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsTurtleManager *) obj)->obj_mutex))
```

ags_turtle_manager_find ()

```
GObject~*
ags_turtle_manager_find (AgsTurtleManager *turtle_manager,
                         gchar *filename);
```

Find *filename* in *turtle_manager* .

Parameters

turtle_manager	the AgsTurtleManager	
filename	the filename as string	

Returns

the matching AgsTurtle, or NULL.

[transfer full]

Since: 3.0.0

ags_turtle_manager_add ()

```
void
ags_turtle_manager_add (AgsTurtleManager *turtle_manager,
                        GObject *turtle);
```

Adds *turtle* to *turtle_manager* .

Parameters

turtle_manager	the AgsTurtleManager	
turtle	the AgsTurtle.	*[transfer full]*

Since: 3.0.0

ags_turtle_manager_get_instance ()

```
AgsTurtleManager~*
ags_turtle_manager_get_instance ();
```

Get instance of AgsTurtleManager.

Returns

the AgsTurtleManager.

[transfer none]

Since: 3.0.0

ags_turtle_manager_new ()

```
AgsTurtleManager~*
ags_turtle_manager_new ();
```

Creates an AgsTurtleManager

Returns

a new AgsTurtleManager

Since: 3.0.0

AGS_IS_TURTLE_MANAGER()

```
#define AGS_IS_TURTLE_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_TURTLE_MANAGER))
```

AGS_IS_TURTLE_MANAGER_CLASS()

```
#define AGS_IS_TURTLE_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_TURTLE_MANAGER))
```

AGS_TURTLE_MANAGER()

```
#define AGS_TURTLE_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TURTLE_MANAGER, AgsTurtleManager))
```

AGS_TURTLE_MANAGER_CLASS()

```
#define AGS_TURTLE_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_TURTLE_MANAGER, AgsTurtleManagerClass))
```

AGS_TURTLE_MANAGER_GET_CLASS()

```
#define AGS_TURTLE_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_TURTLE_MANAGER, AgsTurtleManagerClass))
```

ags_turtle_manager_get_type ()

```
GType
ags_turtle_manager_get_type (void);
```

Types and Values

AGS_TYPE_TURTLE_MANAGER

```
#define AGS_TYPE_TURTLE_MANAGER                 (ags_turtle_manager_get_type())
```

struct AgsTurtleManager

```
struct AgsTurtleManager;
```

struct AgsTurtleManagerClass

```
struct AgsTurtleManagerClass {
  GObjectClass gobject;
};
```

Chapter 16

AgsUUID

AgsUUID — Boxed type of uuid

Stability Level

Stable, unless otherwise indicated

Functions

AgsUUID *	ags_uuid_alloc ()
gpointer	ags_uuid_copy ()
void	ags_uuid_free ()
void	ags_uuid_generate ()
gint	ags_uuid_compare ()
gchar *	ags_uuid_to_string ()
AgsUUID *	ags_uuid_from_string ()
GType	ags_uuid_get_type ()

Types and Values

#define	AGS_UUID_DEFAULT_LENGTH
#define	AGS_UUID_STRING_DEFAULT_LENGTH
#define	AGS_TYPE_UUID
struct	AgsUUID

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsUUID
```

Includes

```
#include <ags/lib/ags_uuid.h>
```

Description

Boxed type of uuid_t data type.

Functions

ags_uuid_alloc ()

```
AgsUUID~*
ags_uuid_alloc ();
```

Allocate AgsUUID.

Returns

a new AgsUUID

Since: 3.0.0

ags_uuid_copy ()

```
gpointer
ags_uuid_copy (AgsUUID *ptr);
```

Create a copy of *ptr* .

Parameters

ptr	the original AgsUUID

Returns

a pointer of the new AgsUUID

Since: 3.0.0

ags_uuid_free ()

```
void
ags_uuid_free (AgsUUID *ptr);
```

Free the memory of *ptr* .

Parameters

ptr	the AgsUUID

Since: 3.0.0

ags_uuid_generate ()

```
void
ags_uuid_generate (AgsUUID *ptr);
```

Generate random UUID.

Parameters

ptr	the AgsUUID

Since: 3.0.0

ags_uuid_compare ()

```
gint
ags_uuid_compare (AgsUUID *a,
                  AgsUUID *b);
```

Compare *a* with *b* .

Parameters

a	an AgsUUID
b	another AgsUUID

Returns

0 on success, otherwise a value less or greater than 0

Since: 3.0.0

ags_uuid_to_string ()

```
gchar~*
ags_uuid_to_string (AgsUUID *ptr);
```

Get string representation of *ptr* .

Parameters

ptr	the AgsUUID

Returns

the UUID as string if *ptr* not NULL, else NULL.

[transfer full]

Since: 3.0.0

ags_uuid_from_string ()

```
AgsUUID~*
ags_uuid_from_string (gchar *str);
```

Retrieve AgsUUID by parsing *str* .

Parameters

str	the UUID as string

Returns

the new AgsUUID on success, otherwise NULL

Since: 3.0.0

ags_uuid_get_type ()

```
GType
ags_uuid_get_type (void);
```

Types and Values

AGS_UUID_DEFAULT_LENGTH

```
#define AGS_UUID_DEFAULT_LENGTH (16)
```

AGS_UUID_STRING_DEFAULT_LENGTH

```
#define AGS_UUID_STRING_DEFAULT_LENGTH (36)
```

AGS_TYPE_UUID

```
#define AGS_TYPE_UUID                (ags_uuid_get_type())
```

struct AgsUUID

```
struct AgsUUID {
  guchar data[16];
};
```

Part II

Util - Helper functions

Chapter 17

AgsDestroyUtil

AgsDestroyUtil — destroy util

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_destroy_util_dispose_and_unref ()
GType	ags_destroy_util_get_type ()

Types and Values

#define	AGS_TYPE_DESTROY_UTIL
struct	AgsDestroyUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsDestroyUtil
```

Includes

```
#include <ags/util/ags_destroy_util.h>
```

Description

Destroying items utility.

Functions

ags_destroy_util_dispose_and_unref ()

```
void
ags_destroy_util_dispose_and_unref (GObject *gobject);
```

Run dispose and unref *gobject* .

Parameters

gobject	the GObject to destroy.	*[transfer full]*

Since: 3.0.0

ags_destroy_util_get_type ()

```
GType
ags_destroy_util_get_type (void);
```

Types and Values

AGS_TYPE_DESTROY_UTIL

```
#define AGS_TYPE_DESTROY_UTIL          (ags_destroy_util_get_type())
```

struct AgsDestroyUtil

```
struct AgsDestroyUtil {
  //empty
};
```

Chapter 18

AgsIdGenerator

AgsIdGenerator — UUID util functions

Stability Level

Stable, unless otherwise indicated

Functions

gchar *	ags_id_generator_create_uuid ()

Includes

```
#include <ags/util/ags_id_generator.h>
```

Description

Functions related to UUID.

Functions

ags_id_generator_create_uuid ()

```
gchar~*
ags_id_generator_create_uuid ();
```

Creates a new string containing an UUID.

Returns

the UUID as string.

[transfer full]

Since: 3.0.0

Chapter 19

AgsListUtil

AgsListUtil — list util

Stability Level

Stable, unless otherwise indicated

Functions

GList *	ags_list_util_find_type ()
GType	ags_list_util_get_type ()

Types and Values

#define	AGS_TYPE_LIST_UTIL
struct	AgsListUtil

Object Hierarchy

```
GBoxed
&#x2570;&#x2500;&#x2500; AgsListUtil
```

Includes

```
#include <ags/util/ags_list_util.h>
```

Description

List utility.

Functions

ags_list_util_find_type ()

```
GList~*
ags_list_util_find_type (GList *list,
                         GType gtype);
```

Find *gtype* within *list* .

Parameters

list	the GList.	*[transfer none][element-type GObject]*
gtype	the GType to find	

Returns

the next matching GList.

[transfer none][element-type GObject]

Since: 3.0.0

ags_list_util_get_type ()

```
GType
ags_list_util_get_type (void);
```

Types and Values

AGS_TYPE_LIST_UTIL

```
#define AGS_TYPE_LIST_UTIL          (ags_list_util_get_type())
```

struct AgsListUtil

```
struct AgsListUtil {
  //empty
};
```

Chapter 20

AgsSoundcardHelper

AgsSoundcardHelper — soundcard helper

Stability Level

Stable, unless otherwise indicated

Functions

guint	ags_soundcard_helper_config_get_dsp_channels ()
guint	ags_soundcard_helper_config_get_pcm_channels ()
gdouble	ags_soundcard_helper_config_get_samplerate ()
guint	ags_soundcard_helper_config_get_buffer_size ()
guint	ags_soundcard_helper_config_get_format ()

Includes

```
#include <ags/util/ags_soundcard_helper.h>
```

Description

Common helper functions related to AgsSoundcard.

Functions

ags_soundcard_helper_config_get_dsp_channels ()

```
guint
ags_soundcard_helper_config_get_dsp_channels
                                (AgsConfig *config);
```

Get dsp channels count.

Parameters

config	the AgsConfig

Returns

the count of dsp channels

Since: 3.0.0

ags_soundcard_helper_config_get_pcm_channels ()

```
guint
ags_soundcard_helper_config_get_pcm_channels
                                (AgsConfig *config);
```

Get pcm channels count.

Parameters

config	the AgsConfig

Returns

the count of pcm channels

Since: 3.0.0

ags_soundcard_helper_config_get_samplerate ()

```
gdouble
ags_soundcard_helper_config_get_samplerate
                                (AgsConfig *config);
```

Get samplerate.

Parameters

config	the AgsConfig

Returns

the samplerate

Since: 3.0.0

ags_soundcard_helper_config_get_buffer_size ()

```
guint
ags_soundcard_helper_config_get_buffer_size
                                (AgsConfig *config);
```

Get buffer size.

Parameters

config	the AgsConfig

Returns

the buffer size

Since: 3.0.0

ags_soundcard_helper_config_get_format ()

```
guint
ags_soundcard_helper_config_get_format
                                (AgsConfig *config);
```

Get format as AgsSoundcardFormat.

Parameters

config	the AgsConfig

Returns

the format

Since: 3.0.0

Part III

Object - Interfaces and marshallers

Chapter 21

AgsApplicable

AgsApplicable — unique access to applicable objects

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_applicable_set_update ()
void	ags_applicable_apply ()
void	ags_applicable_reset ()
#define	AGS_APPLICABLE()
#define	AGS_APPLICABLE_GET_INTERFACE()
#define	AGS_APPLICABLE_INTERFACE()
#define	AGS_IS_APPLICABLE()
#define	AGS_IS_APPLICABLE_INTERFACE()
GType	ags_applicable_get_type ()

Types and Values

#define	AGS_TYPE_APPLICABLE
	AgsApplicable
struct	AgsApplicableInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsApplicable
```

Includes

```
#include <ags/object/ags_applicable.h>
```

Description

The AgsApplicable interface gives you a unique access to applicable objects.

Functions

ags_applicable_set_update ()

```
void
ags_applicable_set_update (AgsApplicable *applicable,
                           gboolean update);
```

Update behaviour.

Parameters

applicable	an AgsApplicable	
update	if TRUE do ::reset after ::apply	

Since: 3.0.0

ags_applicable_apply ()

```
void
ags_applicable_apply (AgsApplicable *applicable);
```

Apply all changes done so far.

Parameters

applicable	an AgsApplicable	

Since: 3.0.0

ags_applicable_reset ()

```
void
ags_applicable_reset (AgsApplicable *applicable);
```

Reset all changes within the user interface.

Parameters

applicable	an AgsApplicable	

Since: 3.0.0

AGS_APPLICABLE()

```
#define AGS_APPLICABLE(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLICABLE, AgsApplicable))
```

AGS_APPLICABLE_GET_INTERFACE()

```
#define AGS_APPLICABLE_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_APPLICABLE, AgsApplicableInterface))
```

AGS_APPLICABLE_INTERFACE()

```
#define AGS_APPLICABLE_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_APPLICABLE, AgsApplicableInterface))
```

AGS_IS_APPLICABLE()

```
#define AGS_IS_APPLICABLE(obj)                  (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_APPLICABLE))
```

AGS_IS_APPLICABLE_INTERFACE()

```
#define AGS_IS_APPLICABLE_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_APPLICABLE))
```

ags_applicable_get_type ()

```
GType
ags_applicable_get_type ();
```

Types and Values

AGS_TYPE_APPLICABLE

```
#define AGS_TYPE_APPLICABLE                     (ags_applicable_get_type())
```

AgsApplicable

```
typedef struct _AgsApplicable AgsApplicable;
```

struct AgsApplicableInterface

```
struct AgsApplicableInterface {
  GTypeInterface ginterface;

  void (*set_update)(AgsApplicable *applicable, gboolean update);
  void (*apply)(AgsApplicable *applicable);
  void (*reset)(AgsApplicable *applicable);
};
```

Chapter 22

AgsApplicationContext

AgsApplicationContext — The application context

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_APPLICATION_CONTEXT_GET_OBJ_MUTEX()
gboolean	ags_application_context_test_flags ()
void	ags_application_context_set_flags ()
void	ags_application_context_unset_flags ()
void	ags_application_context_load_config ()
void	ags_application_context_prepare ()
void	ags_application_context_setup ()
void	ags_application_context_register_types ()
void	ags_application_context_quit ()
AgsApplicationContext *	ags_application_context_get_instance ()
AgsApplicationContext *	ags_application_context_new ()
#define	AGS_APPLICATION_CONTEXT()
#define	AGS_APPLICATION_CONTEXT_CLASS()
#define	AGS_APPLICATION_CONTEXT_GET_CLASS()
#define	AGS_IS_APPLICATION_CONTEXT()
#define	AGS_IS_APPLICATION_CONTEXT_CLASS()
GType	ags_application_context_flags_get_type ()
GType	ags_application_context_get_type ()

Properties

GObject *	config	Read / Write
GObject *	file	Read / Write
GObject *	main-loop	Read / Write
GObject *	task-launcher	Read / Write

Signals

void	load-config	Run Last
void	prepare	Run Last
void	quit	Run Last
void	register-types	Run Last
void	setup	Run Last

Types and Values

#define	AGS_VERSION
#define	AGS_BUILD_ID
#define	AGS_DEFAULT_DIRECTORY
#define	AGS_DEFAULT_CONFIG
enum	AgsApplicationContextFlags
#define	AGS_TYPE_APPLICATION_CONTEXT
#define	AGS_TYPE_APPLICATION_CONTEXT_FLAGS
struct	AgsApplicationContext
struct	AgsApplicationContextClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsApplicationContextFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsApplicationContext
        &#x251c;&#x2500;&#x2500; AgsServerApplicationContext
        &#x2570;&#x2500;&#x2500; AgsThreadApplicationContext
```

Implemented Interfaces

AgsApplicationContext implements AgsConnectable.

Includes

```
#include <ags/object/ags_application_context.h>
```

Description

AgsApplicationContext is a context provider is your and libraries entry point to the application. You might subtype it to implement your own contices. Thus you should consider to create a provider interface for reusability.

Functions

AGS_APPLICATION_CONTEXT_GET_OBJ_MUTEX()

```
#define AGS_APPLICATION_CONTEXT_GET_OBJ_MUTEX(obj) (&(((AgsApplicationContext *) obj)-> ↩
    obj_mutex))
```

ags_application_context_test_flags ()

```
gboolean
ags_application_context_test_flags (AgsApplicationContext *application_context,
                                    guint flags);
```

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

Parameters

application_context	the AgsApplicationContext	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_application_context_set_flags ()

```
void
ags_application_context_set_flags (AgsApplicationContext *application_context,
                                   guint flags);
```

Enable a feature of AgsApplicationContext.

Parameters

application_context	the AgsApplicationContext	
flags	see enum AgsApplication-ContextFlags	

Since: 3.0.0

ags_application_context_unset_flags ()

```
void
ags_application_context_unset_flags (AgsApplicationContext *application_context,
                                     guint flags);
```

Disable a feature of AgsApplicationContext.

Parameters

application_context	the AgsApplicationContext	
flags	see enum AgsApplication-ContextFlags	

Since: 3.0.0

ags_application_context_load_config ()

```
void
ags_application_context_load_config (AgsApplicationContext *application_context);
```

Signal to load and parse configuration.

Parameters

application_context	the AgsApplicationContext	

Since: 3.0.0

ags_application_context_prepare ()

```
void
ags_application_context_prepare (AgsApplicationContext *application_context);
```

Prepare *application_context* .

Parameters

application_context	the AgsApplicationContext	

Since: 3.0.0

ags_application_context_setup ()

```
void
ags_application_context_setup (AgsApplicationContext *application_context);
```

Setup *application_context* .

Parameters

application_context	the AgsApplicationContext	

Since: 3.0.0

ags_application_context_register_types ()

```
void
ags_application_context_register_types
                                (AgsApplicationContext *application_context);
```

Notification to register your types.

Parameters

application_context	the AgsApplicationContext

Since: 3.0.0

ags_application_context_quit ()

```
void
ags_application_context_quit (AgsApplicationContext *application_context);
```

Calls exit()

Parameters

application_context	the AgsApplicationContext

Since: 3.0.0

ags_application_context_get_instance ()

```
AgsApplicationContext~*
ags_application_context_get_instance ();
```

Get your application context instance.

Returns

the AgsApplicationContext instance.

[transfer none]

Since: 3.0.0

ags_application_context_new ()

```
AgsApplicationContext~*
ags_application_context_new (GObject *main_loop,
                             AgsConfig *config);
```

Create a new instance of AgsApplicationContext

Parameters

main_loop	the AgsMainLoop.	*[nullable]*
config	the AgsConfig.	*[nullable]*

Returns

the AgsApplicationContext instance

Since: 3.0.0

AGS_APPLICATION_CONTEXT()

```
#define AGS_APPLICATION_CONTEXT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_APPLICATION_CONTEXT, AgsApplicationContext))
```

AGS_APPLICATION_CONTEXT_CLASS()

```
#define AGS_APPLICATION_CONTEXT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_APPLICATION_CONTEXT, AgsApplicationContextClass))
```

AGS_APPLICATION_CONTEXT_GET_CLASS()

```
#define AGS_APPLICATION_CONTEXT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_APPLICATION_CONTEXT, AgsApplicationContextClass))
```

AGS_IS_APPLICATION_CONTEXT()

```
#define AGS_IS_APPLICATION_CONTEXT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_APPLICATION_CONTEXT))
```

AGS_IS_APPLICATION_CONTEXT_CLASS()

```
#define AGS_IS_APPLICATION_CONTEXT_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_APPLICATION_CONTEXT))
```

ags_application_context_flags_get_type ()

```
GType
ags_application_context_flags_get_type
                                ();
```

ags_application_context_get_type ()

```
GType
ags_application_context_get_type ();
```

Types and Values

AGS_VERSION

```
#define AGS_VERSION "3.0.0"
```

AGS_BUILD_ID

```
#define AGS_BUILD_ID "Wed Nov  6 18:28:15 UTC 2019"
```

AGS_DEFAULT_DIRECTORY

```
#define AGS_DEFAULT_DIRECTORY ".gsequencer"
```

AGS_DEFAULT_CONFIG

```
#define AGS_DEFAULT_CONFIG "ags.conf"
```

enum AgsApplicationContextFlags

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

Members

AGS_APPLICATION_CONTEXT_TYPES_REGISTERED	indicates the types have been registered

AGS_TYPE_APPLICATION_CONTEXT

```
#define AGS_TYPE_APPLICATION_CONTEXT                (ags_application_context_get_type())
```

AGS_TYPE_APPLICATION_CONTEXT_FLAGS

```
#define AGS_TYPE_APPLICATION_CONTEXT_FLAGS          (ags_application_context_flags_get_type ↩
    ())
```

struct AgsApplicationContext

```
struct AgsApplicationContext;
```

struct AgsApplicationContextClass

```
struct AgsApplicationContextClass {
  GObjectClass gobject;

  void (*load_config)(AgsApplicationContext *application_context);

  void (*prepare)(AgsApplicationContext *application_context);
  void (*setup)(AgsApplicationContext *application_context);

  void (*register_types)(AgsApplicationContext *application_context);

  void (*read)(GObject *file, xmlNode *node, GObject **gobject);
  xmlNode* (*write)(GObject *file, xmlNode *parent, GObject *gobject);

  void (*quit)(AgsApplicationContext *application_context);
};
```

Property Details

The "config" property

```
  "config"                    GObject~*
```

The assigned config.

Owner: AgsApplicationContext

Flags: Read / Write

Since: 3.0.0

The "file" property

```
  "file"                      GObject~*
```

The assigned file.

Owner: AgsApplicationContext

Flags: Read / Write

Since: 3.0.0

The "main-loop" property

```
  "main-loop"                 GObject~*
```

The assigned main-loop.

Owner: AgsApplicationContext

Flags: Read / Write

Since: 3.0.0

The "task-launcher" property

```
  "task-launcher"            GObject~*
```

The assigned task launcher.

Owner: AgsApplicationContext

Flags: Read / Write

Since: 3.0.0

Signal Details

The "load-config" signal

```
void
user_function (AgsApplicationContext *application_context,
               gpointer               user_data)
```

The ::load-config notifies to load configuration.

Parameters

application_context	the AgsApplicationContext	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "prepare" signal

```
void
user_function (AgsApplicationContext *application_context,
               gpointer               user_data)
```

The ::prepare signal should be implemented to prepare your application context.

Parameters

application_context	the AgsApplicationContext	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "quit" signal

```
void
user_function (AgsApplicationContext *application_context,
               gpointer               user_data)
```

The ::quit notifies to load configuration.

Parameters

application_context	the AgsApplicationContext	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "register-types" signal

```
void
user_function (AgsApplicationContext *application_context,
               gpointer               user_data)
```

The ::register-types signal should be implemented to load your types.

Parameters

application_context	the AgsApplicationContext	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "setup" signal

```
void
user_function (AgsApplicationContext *application_context,
               gpointer               user_data)
```

The ::setup signal should be implemented to setup your application context.

Parameters

application_context	the AgsApplicationContext	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 23

AgsConfig

AgsConfig — Config Advanced Gtk+ Sequencer

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CONFIG_GET_OBJ_MUTEX()
void	ags_config_load_defaults ()
void	ags_config_load_from_file ()
void	ags_config_load_from_data ()
void	ags_config_set_value ()
gchar *	ags_config_get_value ()
void	ags_config_to_data ()
void	ags_config_save ()
void	ags_config_clear ()
AgsConfig *	ags_config_get_instance ()
AgsConfig *	ags_config_new ()
#define	AGS_CONFIG()
#define	AGS_CONFIG_CLASS()
#define	AGS_CONFIG_GET_CLASS()
#define	AGS_IS_CONFIG()
#define	AGS_IS_CONFIG_CLASS()
GType	ags_config_get_type ()

Signals

char*	get-value	Run Last
void	load-defaults	Run Last
void	set-value	Run Last

Types and Values

#define	AGS_CONFIG_DEFAULT_VERSION
#define	AGS_CONFIG_DEFAULT_BUILD_ID
#define	AGS_CONFIG_GENERIC
#define	AGS_CONFIG_THREAD
#define	AGS_CONFIG_SERVER
#define	AGS_CONFIG_SOUNDCARD
#define	AGS_CONFIG_SOUNDCARD_0
#define	AGS_CONFIG_SEQUENCER
#define	AGS_CONFIG_SEQUENCER_0
#define	AGS_CONFIG_RECALL
#define	AGS_CONFIG_OSC_SERVER
#define	AGS_CONFIG_OSC_SERVER_0
#define	AGS_TYPE_CONFIG
struct	AgsConfig
struct	AgsConfigClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsConfig
```

Includes

```
#include <ags/object/ags_config.h>
```

Description

AgsConfig provides configuration to Advanced Gtk+ Sequencer.

Functions

AGS_CONFIG_GET_OBJ_MUTEX()

```
#define AGS_CONFIG_GET_OBJ_MUTEX(obj) (&(((AgsConfig *) obj)->obj_mutex))
```

ags_config_load_defaults ()

```
void
ags_config_load_defaults (AgsConfig *config);
```

Load configuration from default values.

Parameters

config	the AgsConfig	

Since: 3.0.0

ags_config_load_from_file ()

```
void
ags_config_load_from_file (AgsConfig *config,
                           gchar *filename);
```

Load configuration from *filename* .

Parameters

config	the AgsConfig	
filename	the configuration file	

Since: 3.0.0

ags_config_load_from_data ()

```
void
ags_config_load_from_data (AgsConfig *config,
                           char *buffer,
                           gsize buffer_length);
```

Read configuration in memory.

Parameters

config	the AgsConfig	
buffer	the data buffer	
buffer_length	the size of the buffer	

Since: 3.0.0

ags_config_set_value ()

```
void
ags_config_set_value (AgsConfig *config,
                      gchar *group,
                      gchar *key,
                      gchar *value);
```

Set config by *group* and *key* , applying *value* .

Parameters

config	the AgsConfig	
group	the config group identifier	
key	the key of the property	
value	the value to set	

Since: 3.0.0

ags_config_get_value ()

```
gchar~*
ags_config_get_value (AgsConfig *config,
                      gchar *group,
                      gchar *key);
```

Retrieve config by *group* and *key* .

Parameters

config	the AgsConfig	
group	the config group identifier	
key	the key of the property	

Returns

the property's value.

[transfer full]

Since: 3.0.0

ags_config_to_data ()

```
void
ags_config_to_data (AgsConfig *config,
                    char **buffer,
                    gsize *buffer_length);
```

Save configuration.

Parameters

config	the AgsConfig	
buffer	the data buffer	
buffer_length	the size of the buffer	

Since: 3.0.0

ags_config_save ()

```
void
ags_config_save (AgsConfig *config);
```

Save configuration.

Parameters

config	the AgsConfig

Since: 3.0.0

ags_config_clear ()

```
void
ags_config_clear (AgsConfig *config);
```

Clears configuration.

Parameters

config	the AgsConfig

Since: 3.0.0

ags_config_get_instance ()

```
AgsConfig~*
ags_config_get_instance ();
```

Get config instance.

Returns

the config instance.

[transfer none]

Since: 3.0.0

ags_config_new ()

```
AgsConfig~*
ags_config_new ();
```

Create a new instance of AgsConfig.

Returns

the new AgsConfig.

Since: 3.0.0

AGS_CONFIG()

```
#define AGS_CONFIG(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_CONFIG,  ↩
    AgsConfig))
```

AGS_CONFIG_CLASS()

```
#define AGS_CONFIG_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_CONFIG,  ↩
    AgsConfigClass))
```

AGS_CONFIG_GET_CLASS()

```
#define AGS_CONFIG_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_CONFIG,  ↩
    AgsConfigClass))
```

AGS_IS_CONFIG()

```
#define AGS_IS_CONFIG(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_CONFIG))
```

AGS_IS_CONFIG_CLASS()

```
#define AGS_IS_CONFIG_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_CONFIG))
```

ags_config_get_type ()

```
GType
ags_config_get_type ();
```

Types and Values

AGS_CONFIG_DEFAULT_VERSION

```
#define AGS_CONFIG_DEFAULT_VERSION "0.7.0"
```

AGS_CONFIG_DEFAULT_BUILD_ID

```
#define AGS_CONFIG_DEFAULT_BUILD_ID "CEST 13-10-2015 01:19"
```

AGS_CONFIG_GENERIC

```
#define AGS_CONFIG_GENERIC "generic"
```

AGS_CONFIG_THREAD

```
#define AGS_CONFIG_THREAD "thread"
```

AGS_CONFIG_SERVER

```
#define AGS_CONFIG_SERVER "server"
```

AGS_CONFIG_SOUNDCARD

```
#define AGS_CONFIG_SOUNDCARD "soundcard"
```

AGS_CONFIG_SOUNDCARD_0

```
#define AGS_CONFIG_SOUNDCARD_0 "soundcard-0"
```

AGS_CONFIG_SEQUENCER

```
#define AGS_CONFIG_SEQUENCER "sequencer"
```

AGS_CONFIG_SEQUENCER_0

```
#define AGS_CONFIG_SEQUENCER_0 "sequencer-0"
```

AGS_CONFIG_RECALL

```
#define AGS_CONFIG_RECALL "recall"
```

AGS_CONFIG_OSC_SERVER

```
#define AGS_CONFIG_OSC_SERVER "osc-server"
```

AGS_CONFIG_OSC_SERVER_0

```
#define AGS_CONFIG_OSC_SERVER_0 "osc-server-0"
```

AGS_TYPE_CONFIG

```
#define AGS_TYPE_CONFIG                 (ags_config_get_type ())
```

struct AgsConfig

```
struct AgsConfig;
```

struct AgsConfigClass

```
struct AgsConfigClass {
  GObjectClass gobject;

  void (*load_defaults)(AgsConfig *config);

  void (*set_value)(AgsConfig *config, gchar *group, gchar *key, gchar *value);
  gchar* (*get_value)(AgsConfig *config, gchar *group, gchar *key);
};
```

Signal Details

The "get-value" signal

```
char*
user_function (AgsConfig *config,
               char      *group,
               char      *key,
               gpointer   user_data)
```

The ::get-value signal notifies about value been getting.

Parameters

config	the object to resolve	
group	the group to retrieve from	
key	the key to get	
user_data	user data set when the signal handler was connected.	

Returns

the value

Flags: Run Last

Since: 3.0.0

The "load-defaults" signal

```
void
user_function (AgsConfig *config,
               gpointer   user_data)
```

The ::load-defaults signal notifies about loading defaults

Parameters

config	the object to resolve	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "set-value" signal

```
void
user_function (AgsConfig *config,
               char      *group,
               char      *key,
               char      *value,
               gpointer   user_data)
```

The ::set-value signal notifies about value been setting.

Parameters

config	the object to resolve	
group	the group to apply to	
key	the key to set	
value	the value to apply	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 24

AgsPriority

AgsPriority — Priority Advanced Gtk+ Sequencer

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PRIORITY_GET_OBJ_MUTEX()
void	ags_priority_load_defaults ()
void	ags_priority_load_from_file ()
void	ags_priority_set_value ()
gchar *	ags_priority_get_value ()
AgsPriority *	ags_priority_get_instance ()
AgsPriority *	ags_priority_new ()
#define	AGS_IS_PRIORITY()
#define	AGS_IS_PRIORITY_CLASS()
#define	AGS_PRIORITY()
#define	AGS_PRIORITY_CLASS()
#define	AGS_PRIORITY_GET_CLASS()
GType	ags_priority_get_type ()

Signals

char*	get-value	Run Last
void	load-defaults	Run Last
void	set-value	Run Last

Types and Values

#define	AGS_TYPE_PRIORITY_FLAGS
#define	AGS_PRIORITY_DEFAULT_VERSION

#define	AGS_PRIORITY_DEFAULT_BUILD_ID
#define	AGS_PRIORITY_RT_THREAD
#define	AGS_PRIORITY_KEY_LIBAGS
#define	AGS_PRIORITY_KEY_SERVER_MAIN_LOOP
#define	AGS_PRIORITY_KEY_AUDIO_MAIN_LOOP
#define	AGS_PRIORITY_KEY_AUDIO
#define	AGS_PRIORITY_KEY_OSC_SERVER_MAIN_LOOP
#define	AGS_PRIORITY_KEY_GUI_MAIN_LOOP
#define	AGS_TYPE_PRIORITY
struct	AgsPriority
struct	AgsPriorityClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsPriority
```

Includes

```
#include <ags/object/ags_priority.h>
```

Description

AgsPriority provides priorities to Advanced Gtk+ Sequencer.

Functions

AGS_PRIORITY_GET_OBJ_MUTEX()

```
#define AGS_PRIORITY_GET_OBJ_MUTEX(obj) (&(((AgsPriority *) obj)->obj_mutex))
```

ags_priority_load_defaults ()

```
void
ags_priority_load_defaults (AgsPriority *priority);
```

Load priorities from default values.

Parameters

priority	the AgsPriority

Since: 3.0.0

ags_priority_load_from_file ()

```
void
ags_priority_load_from_file (AgsPriority *priority,
                             gchar *filename);
```

Load priorities from *filename* .

Parameters

priority	the AgsPriority	
filename	the priorityuration file	

Since: 3.0.0

ags_priority_set_value ()

```
void
ags_priority_set_value (AgsPriority *priority,
                        gchar *group,
                        gchar *key,
                        gchar *value);
```

Set priority by *group* and *key* , applying *value* .

Parameters

priority	the AgsPriority	
group	the priority group identifier	
key	the key of the property	
value	the value to set	

Since: 3.0.0

ags_priority_get_value ()

```
gchar~*
ags_priority_get_value (AgsPriority *priority,
                        gchar *group,
                        gchar *key);
```

Retrieve priority by *group* and *key* .

Parameters

priority	the AgsPriority	
group	the priority group identifier	
key	the key of the property	

Returns

the property's value.

[transfer full]

Since: 3.0.0

ags_priority_get_instance ()

```
AgsPriority~*
ags_priority_get_instance ();
```

Get priority instance.

Returns

the priority instance.

[transfer none]

Since: 3.0.0

ags_priority_new ()

```
AgsPriority~*
ags_priority_new ();
```

Create a new instance of AgsPriority.

Returns

the new AgsPriority.

Since: 3.0.0

AGS_IS_PRIORITY()

```
#define AGS_IS_PRIORITY(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_PRIORITY))
```

AGS_IS_PRIORITY_CLASS()

```
#define AGS_IS_PRIORITY_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class),  ↩
    AGS_TYPE_PRIORITY))
```

AGS_PRIORITY()

```
#define AGS_PRIORITY(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PRIORITY, AgsPriority))
```

AGS_PRIORITY_CLASS()

```
#define AGS_PRIORITY_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_PRIORITY, AgsPriorityClass))
```

AGS_PRIORITY_GET_CLASS()

```
#define AGS_PRIORITY_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS((obj),  ↩
    AGS_TYPE_PRIORITY, AgsPriorityClass))
```

ags_priority_get_type ()

```
GType
ags_priority_get_type ();
```

Types and Values

AGS_TYPE_PRIORITY_FLAGS

```
#define AGS_TYPE_PRIORITY_FLAGS          (ags_priority_flags_get_type())
```

AGS_PRIORITY_DEFAULT_VERSION

```
#define AGS_PRIORITY_DEFAULT_VERSION "2.4.2"
```

AGS_PRIORITY_DEFAULT_BUILD_ID

```
#define AGS_PRIORITY_DEFAULT_BUILD_ID "Mon Dec  2 08:15:02 UTC 2019"
```

AGS_PRIORITY_RT_THREAD

```
#define AGS_PRIORITY_RT_THREAD "rt-thread"
```

AGS_PRIORITY_KEY_LIBAGS

```
#define AGS_PRIORITY_KEY_LIBAGS "libags"
```

AGS_PRIORITY_KEY_SERVER_MAIN_LOOP

```
#define AGS_PRIORITY_KEY_SERVER_MAIN_LOOP "server-main-loop"
```

AGS_PRIORITY_KEY_AUDIO_MAIN_LOOP

```
#define AGS_PRIORITY_KEY_AUDIO_MAIN_LOOP "audio-main-loop"
```

AGS_PRIORITY_KEY_AUDIO

```
#define AGS_PRIORITY_KEY_AUDIO "audio"
```

AGS_PRIORITY_KEY_OSC_SERVER_MAIN_LOOP

```
#define AGS_PRIORITY_KEY_OSC_SERVER_MAIN_LOOP "osc-server-main-loop"
```

AGS_PRIORITY_KEY_GUI_MAIN_LOOP

```
#define AGS_PRIORITY_KEY_GUI_MAIN_LOOP "gui-main-loop"
```

AGS_TYPE_PRIORITY

```
#define AGS_TYPE_PRIORITY                (ags_priority_get_type ())
```

struct AgsPriority

```
struct AgsPriority;
```

struct AgsPriorityClass

```
struct AgsPriorityClass {
  GObjectClass gobject;

  void (*load_defaults)(AgsPriority *priority);

  void (*set_value)(AgsPriority *priority, gchar *group, gchar *key, gchar *value);
  gchar* (*get_value)(AgsPriority *priority, gchar *group, gchar *key);
};
```

Signal Details

The "get-value" signal

```
char*
user_function (AgsPriority *priority,
               char        *group,
               char        *key,
               gpointer     user_data)
```

The ::get-value signal notifies about value been getting.

Parameters

priority	the object to resolve	
group	the group to retrieve from	
key	the key to get	
user_data	user data set when the signal handler was connected.	

Returns

the value

Flags: Run Last

Since: 3.0.0

The "load-defaults" signal

```
void
user_function (AgsPriority *priority,
               gpointer     user_data)
```

The ::load-defaults signal notifies about loading defaults

Parameters

priority	the object to resolve	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "set-value" signal

```
void
user_function (AgsPriority *priority,
               char        *group,
               char        *key,
               char        *value,
               gpointer     user_data)
```

The ::set-value signal notifies about value been setting.

Parameters

priority	the object to resolve	
group	the group to apply to	
key	the key to set	

value	the value to apply	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 25

AgsConnectable

AgsConnectable — unique access to objects

Stability Level

Stable, unless otherwise indicated

Functions

AgsUUID *	ags_connectable_get_uuid ()
gboolean	ags_connectable_has_resource ()
gboolean	ags_connectable_is_ready ()
void	ags_connectable_add_to_registry ()
void	ags_connectable_remove_from_registry ()
xmlNode *	ags_connectable_list_resource ()
xmlNode *	ags_connectable_xml_compose ()
void	ags_connectable_xml_parse ()
gboolean	ags_connectable_is_connected ()
void	ags_connectable_connect ()
void	ags_connectable_disconnect ()
void	ags_connectable_connect_connection ()
void	ags_connectable_disconnect_connection ()
#define	AGS_CONNECTABLE()
#define	AGS_CONNECTABLE_GET_INTERFACE()
#define	AGS_CONNECTABLE_INTERFACE()
#define	AGS_IS_CONNECTABLE()
#define	AGS_IS_CONNECTABLE_INTERFACE()
GType	ags_connectable_get_type ()

Types and Values

enum	AgsConnectableFlags
#define	AGS_TYPE_CONNECTABLE
	AgsConnectable
struct	AgsConnectableInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsConnectable
```

Known Implementations

AgsConnectable is implemented by AgsApplicationContext, AgsDestroyWorker, AgsGenericMainLoop, AgsReturnableThread, AgsServerApplicationContext, AgsServerStatus, AgsTaskCompletion, AgsTaskLauncher, AgsThread, AgsThreadApplicationContext and AgsWorkerThread.

Includes

```
#include <ags/object/ags_connectable.h>
```

Description

The AgsConnectable interface gives you a unique access to all objects and is responsible to set up signal handlers.

Functions

ags_connectable_get_uuid ()

```
AgsUUID~*
ags_connectable_get_uuid (AgsConnectable *connectable);
```

Get UUID of *connectable* .

Parameters

connectable	the AgsConnectable

Returns

the assigned AgsUUID

Since: 3.0.0

ags_connectable_has_resource ()

```
gboolean
ags_connectable_has_resource (AgsConnectable *connectable);
```

Check the connectable to have resources.

Parameters

connectable	the AgsConnectable

Returns

TRUE if *connectable* can be added to registry, otherwise FALSE.

Since: 3.0.0

ags_connectable_is_ready ()

```
gboolean
ags_connectable_is_ready (AgsConnectable *connectable);
```

Connect the connectable.

Parameters

connectable	the AgsConnectable

Returns

TRUE if is added to registry, otherwise FALSE.

Since: 3.0.0

ags_connectable_add_to_registry ()

```
void
ags_connectable_add_to_registry (AgsConnectable *connectable);
```

Add connectable to registry.

Parameters

connectable	the AgsConnectable

Since: 3.0.0

ags_connectable_remove_from_registry ()

```
void
ags_connectable_remove_from_registry (AgsConnectable *connectable);
```

Remove connectable from registry.

Parameters

connectable	the AgsConnectable

Since: 3.0.0

ags_connectable_list_resource ()

```
xmlNode~*
ags_connectable_list_resource (AgsConnectable *connectable);
```

List resources as an XML element and return it.

Parameters

connectable	the AgsConnectable

Returns

the xmlNode.

[transfer none]

Since: 3.0.0

ags_connectable_xml_compose ()

```
xmlNode~*
ags_connectable_xml_compose (AgsConnectable *connectable);
```

Compose an XML element and return it.

Parameters

connectable	the AgsConnectable

Returns

the xmlNode.

[transfer none]

Since: 3.0.0

ags_connectable_xml_parse ()

```
void
ags_connectable_xml_parse (AgsConnectable *connectable,
                           xmlNode *node);
```

Parse `node` as XML element and apply it.

Parameters

connectable	the AgsConnectable	
node	the xmlNode.	*[transfer none]*

Since: 3.0.0

ags_connectable_is_connected ()

```
gboolean
ags_connectable_is_connected (AgsConnectable *connectable);
```

Check if the *connectable* was connected.

Parameters

connectable	the AgsConnectable	

Returns

TRUE if is connected, otherwise FALSE.

Since: 3.0.0

ags_connectable_connect ()

```
void
ags_connectable_connect (AgsConnectable *connectable);
```

Connect the connectable.

Parameters

connectable	the AgsConnectable	

Since: 3.0.0

ags_connectable_disconnect ()

```
void
ags_connectable_disconnect (AgsConnectable *connectable);
```

Disconnect the connectable.

Parameters

connectable	the AgsConnectable	

Since: 3.0.0

ags_connectable_connect_connection ()

```
void
ags_connectable_connect_connection (AgsConnectable *connectable,
                                    GObject *connection);
```

Disconnect the connectable.

Parameters

connectable	the AgsConnectable	
connection	the connection	

Since: 3.0.0

ags_connectable_disconnect_connection ()

```
void
ags_connectable_disconnect_connection (AgsConnectable *connectable,
                                       GObject *connection);
```

Disconnect the connectable.

Parameters

connectable	the AgsConnectable	
connection	the connection	

Since: 3.0.0

AGS_CONNECTABLE()

```
#define AGS_CONNECTABLE(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CONNECTABLE, AgsConnectable))
```

AGS_CONNECTABLE_GET_INTERFACE()

```
#define AGS_CONNECTABLE_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_CONNECTABLE, AgsConnectableInterface))
```

AGS_CONNECTABLE_INTERFACE()

```
#define AGS_CONNECTABLE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_CONNECTABLE, AgsConnectableInterface))
```

AGS_IS_CONNECTABLE()

```
#define AGS_IS_CONNECTABLE(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CONNECTABLE))
```

AGS_IS_CONNECTABLE_INTERFACE()

```
#define AGS_IS_CONNECTABLE_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_CONNECTABLE))
```

ags_connectable_get_type ()

```
GType
ags_connectable_get_type ();
```

Types and Values

enum AgsConnectableFlags

Enum values to specify particular state.

Members

AGS_CONNECTABLE_ADDED_TO_REGISTRY	added to registry
AGS_CONNECTABLE_CONNECTED	connected

AGS_TYPE_CONNECTABLE

```
#define AGS_TYPE_CONNECTABLE                    (ags_connectable_get_type())
```

AgsConnectable

```
typedef struct _AgsConnectable AgsConnectable;
```

struct AgsConnectableInterface

```
struct AgsConnectableInterface {
  GTypeInterface ginterface;

  AgsUUID* (*get_uuid)(AgsConnectable *connectable);
  gboolean (*has_resource)(AgsConnectable *connectable);

  gboolean (*is_ready)(AgsConnectable *connectable);
  void (*add_to_registry)(AgsConnectable *connectable);
  void (*remove_from_registry)(AgsConnectable *connectable);

  xmlNode* (*list_resource)(AgsConnectable *connectable);

  xmlNode* (*xml_compose)(AgsConnectable *connectable);
  void (*xml_parse)(AgsConnectable *connectable,
        xmlNode *node);
```

```
  gboolean (*is_connected)(AgsConnectable *connectable);
  void (*connect)(AgsConnectable *connectable);
  void (*disconnect)(AgsConnectable *connectable);

  void (*connect_connection)(AgsConnectable *connectable,
           GObject *connection);
  void (*disconnect_connection)(AgsConnectable *connectable,
        GObject *connection);
};
```

Chapter 26

AgsCountable

AgsCountable — a counter interface

Stability Level

Stable, unless otherwise indicated

Functions

guint64	ags_countable_get_sequencer_counter ()
guint64	ags_countable_get_notation_counter ()
guint64	ags_countable_get_wave_counter ()
guint64	ags_countable_get_midi_counter ()
#define	AGS_COUNTABLE()
#define	AGS_COUNTABLE_GET_INTERFACE()
#define	AGS_COUNTABLE_INTERFACE()
#define	AGS_IS_COUNTABLE()
#define	AGS_IS_COUNTABLE_INTERFACE()
GType	ags_countable_get_type ()

Types and Values

#define	AGS_TYPE_COUNTABLE
	AgsCountable
struct	AgsCountableInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsCountable
```

Includes

```
#include <ags/object/ags_countable.h>
```

Description

The AgsCountable interface gives you a unique access to counters. Generally we distinguish between notation and sequencer counters. Those may have their own relative positions, especially the sequencers may loop.

Functions

ags_countable_get_sequencer_counter ()

```
guint64
ags_countable_get_sequencer_counter (AgsCountable *countable);
```

Retrieve current position of sequencer.

Parameters

countable	an AgsCountable

Returns

the current position

Since: 3.0.0

ags_countable_get_notation_counter ()

```
guint64
ags_countable_get_notation_counter (AgsCountable *countable);
```

Retrieve current position of notation.

Parameters

countable	an AgsCountable

Returns

the current position

Since: 3.0.0

ags_countable_get_wave_counter ()

```
guint64
ags_countable_get_wave_counter (AgsCountable *countable);
```

Retrieve current position of wave.

Parameters

countable	an AgsCountable

Returns

the current position

Since: 3.0.0

ags_countable_get_midi_counter ()

```
guint64
ags_countable_get_midi_counter (AgsCountable *countable);
```

Retrieve current position of MIDI.

Parameters

countable	an AgsCountable

Returns

the current position

Since: 3.0.0

AGS_COUNTABLE()

```
#define AGS_COUNTABLE(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_COUNTABLE, AgsCountable))
```

AGS_COUNTABLE_GET_INTERFACE()

```
#define AGS_COUNTABLE_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_COUNTABLE, AgsCountableInterface))
```

AGS_COUNTABLE_INTERFACE()

```
#define AGS_COUNTABLE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_COUNTABLE, AgsCountableInterface))
```

AGS_IS_COUNTABLE()

```
#define AGS_IS_COUNTABLE(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_COUNTABLE))
```

AGS_IS_COUNTABLE_INTERFACE()

```
#define AGS_IS_COUNTABLE_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_COUNTABLE))
```

ags_countable_get_type ()

```
GType
ags_countable_get_type ();
```

Types and Values

AGS_TYPE_COUNTABLE

```
#define AGS_TYPE_COUNTABLE                     (ags_countable_get_type())
```

AgsCountable

```
typedef struct _AgsCountable AgsCountable;
```

struct AgsCountableInterface

```
struct AgsCountableInterface {
  GTypeInterface ginterface;

  guint64 (*get_sequencer_counter)(AgsCountable *countable);
  guint64 (*get_notation_counter)(AgsCountable *countable);
  guint64 (*get_wave_counter)(AgsCountable *countable);
  guint64 (*get_midi_counter)(AgsCountable *countable);
};
```

Chapter 27

AgsCursor

AgsCursor — unique access to cursor classes

Stability Level

Stable, unless otherwise indicated

Functions

gdouble	ags_cursor_get_default_offset ()
gdouble	ags_cursor_get_tact ()
gdouble	ags_cursor_get_bpm ()
guint	ags_cursor_get_rate ()
gdouble	ags_cursor_get_delay ()
guint64	ags_cursor_get_duration ()
gdouble	ags_cursor_get_delay_counter ()
guint64	ags_cursor_get_offset ()
GList *	ags_cursor_get_prev ()
GList *	ags_cursor_get_next ()
GList *	ags_cursor_get_current_copy ()
GList *	ags_cursor_get_current ()
#define	AGS_CURSOR()
#define	AGS_CURSOR_GET_INTERFACE()
#define	AGS_CURSOR_INTERFACE()
#define	AGS_IS_CURSOR()
#define	AGS_IS_CURSOR_INTERFACE()
GType	ags_cursor_get_type ()

Types and Values

#define	AGS_TYPE_CURSOR
	AgsCursor
struct	AgsCursorInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsCursor
```

Includes

```
#include <ags/object/ags_cursor.h>
```

Description

The AgsCursor interface does cursor position abstraction.

Functions

ags_cursor_get_default_offset ()

```
gdouble
ags_cursor_get_default_offset (AgsCursor *cursor);
```

Get the default offset.

Parameters

cursor	the AgsCursor interface

Returns

the default offset

Since: 3.0.0

ags_cursor_get_tact ()

```
gdouble
ags_cursor_get_tact (AgsCursor *cursor);
```

Get the tact.

Parameters

cursor	the AgsCursor interface

Returns

the tact

Since: 3.0.0

ags_cursor_get_bpm ()

```
gdouble
ags_cursor_get_bpm (AgsCursor *cursor);
```

Get BPM.

Parameters

cursor	the AgsCursor interface

Returns

the BPM

Since: 3.0.0

ags_cursor_get_rate ()

```
guint
ags_cursor_get_rate (AgsCursor *cursor);
```

Get offset counter rate.

Parameters

cursor	the AgsCursor interface

Since: 3.0.0

ags_cursor_get_delay ()

```
gdouble
ags_cursor_get_delay (AgsCursor *cursor);
```

Get delay.

Parameters

cursor	the AgsCursor interface

Returns

the delay

Since: 3.0.0

ags_cursor_get_duration ()

```
guint64
ags_cursor_get_duration (AgsCursor *cursor);
```

Get duration.

Parameters

cursor	the AgsCursor interface

Returns

the duration

Since: 3.0.0

ags_cursor_get_delay_counter ()

```
gdouble
ags_cursor_get_delay_counter (AgsCursor *cursor);
```

Get delay counter.

Parameters

cursor	the AgsCursor interface

Returns

the delay counter

Since: 3.0.0

ags_cursor_get_offset ()

```
guint64
ags_cursor_get_offset (AgsCursor *cursor);
```

Get offset.

Parameters

cursor	the AgsCursor interface

Returns

the offset

Since: 3.0.0

ags_cursor_get_prev ()

```
GList~*
ags_cursor_get_prev (AgsCursor *cursor);
```

Get prev.

Parameters

cursor	the AgsCursor interface

Returns

the GList pointing to previous.

[element-type GObject][transfer full]

Since: 3.0.0

ags_cursor_get_next ()

```
GList~*
ags_cursor_get_next (AgsCursor *cursor);
```

Get next.

Parameters

cursor	the AgsCursor interface

Returns

the GList pointing to next.

[element-type GObject][transfer full]

Since: 3.0.0

ags_cursor_get_current_copy ()

```
GList~*
ags_cursor_get_current_copy (AgsCursor *cursor);
```

Get current as copy.

Parameters

cursor	the AgsCursor interface

Returns

the GList of current as copy.

[element-type GObject][transfer full]

Since: 3.0.0

ags_cursor_get_current ()

```
GList~*
ags_cursor_get_current (AgsCursor *cursor);
```

Get current.

Parameters

cursor	the AgsCursor interface

Returns

the GList pointing to current.

[element-type GObject][transfer none]

Since: 3.0.0

AGS_CURSOR()

```
#define AGS_CURSOR(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CURSOR, AgsCursor))
```

AGS_CURSOR_GET_INTERFACE()

```
#define AGS_CURSOR_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_CURSOR, AgsCursorInterface))
```

AGS_CURSOR_INTERFACE()

```
#define AGS_CURSOR_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_CURSOR, AgsCursorInterface))
```

AGS_IS_CURSOR()

```
#define AGS_IS_CURSOR(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CURSOR))
```

AGS_IS_CURSOR_INTERFACE()

```
#define AGS_IS_CURSOR_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_CURSOR))
```

ags_cursor_get_type ()

```
GType
ags_cursor_get_type ();
```

Types and Values

AGS_TYPE_CURSOR

```
#define AGS_TYPE_CURSOR                     (ags_cursor_get_type())
```

AgsCursor

```
typedef struct _AgsCursor AgsCursor;
```

struct AgsCursorInterface

```
struct AgsCursorInterface {
  GTypeInterface ginterface;

  gdouble (*get_default_offset)(AgsCursor *cursor);

  gdouble (*get_tact)(AgsCursor *cursor);
  gdouble (*get_bpm)(AgsCursor *cursor);

  guint (*get_rate)(AgsCursor *cursor);

  gdouble (*get_delay)(AgsCursor *cursor);
  guint64 (*get_duration)(AgsCursor *cursor);

  gdouble (*get_delay_counter)(AgsCursor *cursor);
  guint64 (*get_offset)(AgsCursor *cursor);

  GList* (*get_prev)(AgsCursor *cursor);
  GList* (*get_next)(AgsCursor *cursor);

  GList* (*get_current_copy)(AgsCursor *cursor);
  GList* (*get_current)(AgsCursor *cursor);
};
```

Chapter 28

ags_globals

ags_globals —

Stability Level

Stable, unless otherwise indicated

Functions

GHashTable *	ags_globals_get_hash_table ()
void	ags_globals_set ()
GValue *	ags_globals_get ()

Description

Functions

ags_globals_get_hash_table ()

```
GHashTable~*
ags_globals_get_hash_table ();
```

Get globals hash table.

Returns

the GHashTable containig globals.

[element-type utf8 GObject.Value][transfer none]

Since: 3.0.0

ags_globals_set ()

```
void
ags_globals_set (gchar *key,
                 GValue *value);
```

Set global key/value.

Parameters

key	the key as string	
value	the GValue	

Since: 3.0.0

ags_globals_get ()

```
GValue~*
ags_globals_get (gchar *key);
```

Get global value for *key* .

Parameters

key	the key as string	

Returns

the GValue if *key* found, otherwise NULL

Since: 3.0.0

Chapter 29

ags_macros

ags_macros —

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_DECLARE_INTERFACE()

Description

Functions

AGS_DECLARE_INTERFACE()

```
#define             AGS_DECLARE_INTERFACE(ModuleObjName, module_obj_name, MODULE, OBJ_NAME, ↩
    PrerequisiteName)
```

Chapter 30

AgsMainLoop

AgsMainLoop — main loop interface

Stability Level

Stable, unless otherwise indicated

Functions

GRecMutex *	ags_main_loop_get_tree_lock ()
void	ags_main_loop_set_syncing ()
gboolean	ags_main_loop_is_syncing ()
void	ags_main_loop_set_critical_region ()
gboolean	ags_main_loop_is_critical_region ()
void	ags_main_loop_inc_queued_critical_region ()
void	ags_main_loop_dec_queued_critical_region ()
guint	ags_main_loop_test_queued_critical_region ()
void	ags_main_loop_change_frequency ()
#define	AGS_IS_MAIN_LOOP()
#define	AGS_IS_MAIN_LOOP_INTERFACE()
#define	AGS_MAIN_LOOP()
#define	AGS_MAIN_LOOP_GET_INTERFACE()
#define	AGS_MAIN_LOOP_INTERFACE()
GType	ags_main_loop_get_type ()

Signals

void	change-frequency	Run Last

Types and Values

#define	AGS_TYPE_MAIN_LOOP
	AgsMainLoop

struct	AgsMainLoopInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsMainLoop
```

Known Implementations

AgsMainLoop is implemented by AgsGenericMainLoop.

Includes

```
#include <ags/object/ags_main_loop.h>
```

Description

The AgsMainLoop interface gives you a unique access to the main loop.

Functions

ags_main_loop_get_tree_lock ()

```
GRecMutex~*
ags_main_loop_get_tree_lock (AgsMainLoop *main_loop);
```

Retrieve the tree mutex.

Parameters

main_loop	the AgsMainLoop

Returns

the mutex.

[transfer none]

Since: 3.0.0

ags_main_loop_set_syncing ()

```
void
ags_main_loop_set_syncing (AgsMainLoop *main_loop,
                           gboolean is_syncing);
```

Set thread tree is syncing.

Parameters

main_loop	the AgsMainLoop	
is_syncing	set TRUE if syncing	

Since: 3.0.0

ags_main_loop_is_syncing ()

```
gboolean
ags_main_loop_is_syncing (AgsMainLoop *main_loop);
```

Check if thread tree is syncing.

Parameters

main_loop	the AgsMainLoop	

Returns

TRUE if sync in progress, otherwise FALSE

Since: 3.0.0

ags_main_loop_set_critical_region ()

```
void
ags_main_loop_set_critical_region (AgsMainLoop *main_loop,
                                   gboolean is_critical_region);
```

Set main loop is in critical region.

Parameters

main_loop	the AgsMainLoop	
is_critical_region	set TRUE if critical region	

Since: 3.0.0

ags_main_loop_is_critical_region ()

```
gboolean
ags_main_loop_is_critical_region (AgsMainLoop *main_loop);
```

Check if main loop is in critical region.

Parameters

main_loop	the AgsMainLoop

Returns

TRUE if sync in progress, otherwise FALSE

Since: 3.0.0

ags_main_loop_inc_queued_critical_region ()

```
void
ags_main_loop_inc_queued_critical_region
                                (AgsMainLoop *main_loop);
```

Increment thread needs access to main loop's critical region field.

Parameters

main_loop	the AgsMainLoop

Since: 3.0.0

ags_main_loop_dec_queued_critical_region ()

```
void
ags_main_loop_dec_queued_critical_region
                                (AgsMainLoop *main_loop);
```

Decrement thread needs access to main loop's critical region field.

Parameters

main_loop	the AgsMainLoop

Since: 3.0.0

ags_main_loop_test_queued_critical_region ()

```
guint
ags_main_loop_test_queued_critical_region
                                (AgsMainLoop *main_loop);
```

Test main loop may enter critical region.

Parameters

main_loop	the AgsMainLoop

Returns

0 if main loop may enter critical region, otherwise not

Since: 3.0.0

ags_main_loop_change_frequency ()

```
void
ags_main_loop_change_frequency (AgsMainLoop *main_loop,
                                gdouble frequency);
```

Change frequency.

Parameters

main_loop	the AgsMainLoop	
frequency	the new frequency	

Since: 3.0.0

AGS_IS_MAIN_LOOP()

```
#define AGS_IS_MAIN_LOOP(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_MAIN_LOOP))
```

AGS_IS_MAIN_LOOP_INTERFACE()

```
#define AGS_IS_MAIN_LOOP_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_MAIN_LOOP))
```

AGS_MAIN_LOOP()

```
#define AGS_MAIN_LOOP(obj)                       (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MAIN_LOOP, AgsMainLoop))
```

AGS_MAIN_LOOP_GET_INTERFACE()

```
#define AGS_MAIN_LOOP_GET_INTERFACE(obj)         (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_MAIN_LOOP, AgsMainLoopInterface))
```

AGS_MAIN_LOOP_INTERFACE()

```
#define AGS_MAIN_LOOP_INTERFACE(vtable)          (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_MAIN_LOOP, AgsMainLoopInterface))
```

ags_main_loop_get_type ()

```
GType
ags_main_loop_get_type ();
```

Types and Values

AGS_TYPE_MAIN_LOOP

```
#define AGS_TYPE_MAIN_LOOP                    (ags_main_loop_get_type())
```

AgsMainLoop

```
typedef struct _AgsMainLoop AgsMainLoop;
```

struct AgsMainLoopInterface

```
struct AgsMainLoopInterface {
  GTypeInterface ginterface;

  GRecMutex* (*get_tree_lock)(AgsMainLoop *main_loop);

  void (*set_syncing)(AgsMainLoop *main_loop, gboolean is_syncing);
  gboolean (*is_syncing)(AgsMainLoop *main_loop);

  void (*set_critical_region)(AgsMainLoop *main_loop, gboolean is_critical_region);
  gboolean (*is_critical_region)(AgsMainLoop *main_loop);

  void (*inc_queued_critical_region)(AgsMainLoop *main_loop);
  void (*dec_queued_critical_region)(AgsMainLoop *main_loop);
  guint (*test_queued_critical_region)(AgsMainLoop *main_loop);

  void (*change_frequency)(AgsMainLoop *main_loop,
         gdouble frequency);
};
```

Signal Details

The "change-frequency" signal

```
void
user_function (AgsMainLoop *main_loop,
               double       frequency,
               gpointer     user_data)
```

Change frequency.

Parameters

main_loop	the AgsMainLoop	
frequency	the new frequency	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 31

ags_marshal

ags_marshal —

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_cclosure_marshal_VOID__UINT64 ()
void	ags_cclosure_marshal_VOID__BOOLEAN_POINTER ()
void	ags_cclosure_marshal_VOID__INT_UINT ()
void	ags_cclosure_marshal_VOID__INT_POINTER ()
void	ags_cclosure_marshal_VOID__INT_UINT_POINTER ()
void	ags_cclosure_marshal_VOID__UINT_BOOLEAN ()
void	ags_cclosure_marshal_VOID__UINT_UINT ()
void	ags_cclosure_marshal_VOID__UINT_UINT_BOOLEAN ()
void	ags_cclosure_marshal_VOID__UINT_UINT_UINT ()
void	ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT ()
void	ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT ()
void	ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT_UINT ()
void	ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT_UINT_UINT ()
void	ags_cclosure_marshal_VOID__UINT_DOUBLE_UINT ()
void	ags_cclosure_marshal_VOID__UINT_INT ()
void	ags_cclosure_marshal_VOID__UINT_DOUBLE ()
void	ags_cclosure_marshal_VOID__UINT_STRING_UINT ()
void	ags_cclosure_marshal_VOID__UINT_POINTER_UINT ()
void	ags_cclosure_marshal_VOID__INT64_UINT ()
void	ags_cclosure_marshal_VOID__POINTER_UINT ()
void	ags_cclosure_marshal_VOID__POINTER_UINT_BOOLEAN ()
void	ags_cclosure_marshal_VOID__DOUBLE_DOUBLE ()
void	ags_cclosure_marshal_VOID__POINTER_UINT_UINT ()
void	ags_cclosure_marshal_VOID__STRING_UINT ()
void	ags_cclosure_marshal_VOID__STRING_UINT_DOUBLE ()
void	ags_cclosure_marshal_VOID__STRING_UINT_POINTER ()
void	ags_cclosure_marshal_VOID__STRING_UINT_STRING_DOUBLE ()

void	ags_cclosure_marshal_VOID__STRING_POINTER ()
void	ags_cclosure_marshal_VOID__STRING_STRING_STRING ()
void	ags_cclosure_marshal_VOID__POINTER_INT ()
void	ags_cclosure_marshal_VOID__POINTER_INT_POINTER ()
void	ags_cclosure_marshal_VOID__POINTER_INT_INT_BOOLEAN ()
void	ags_cclosure_marshal_VOID__POINTER_POINTER ()
void	ags_cclosure_marshal_VOID__POINTER_UINT_POINTER ()
void	ags_cclosure_marshal_VOID__POINTER_POINTER_UINT ()
void	ags_cclosure_marshal_VOID__POINTER_POINTER_POINTER ()
void	ags_cclosure_marshal_VOID__POINTER_STRING_STRING_STRING_INT ()
void	ags_cclosure_marshal_VOID__POINTER_OBJECT_OBJECT_STRING_STRING_STRIN ()
void	ags_cclosure_marshal_VOID__OBJECT_UINT ()
void	ags_cclosure_marshal_VOID__OBJECT_UINT_UINT_UINT ()
void	ags_cclosure_marshal_VOID__OBJECT_OBJECT ()
void	ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_INT_BOOLEAN ()
void	ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJEC ()
void	ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJEC ()
void	ags_cclosure_marshal_POINTER__VOID ()
void	ags_cclosure_marshal_INT__VOID ()
void	ags_cclosure_marshal_INT64__OBJECT ()
void	ags_cclosure_marshal_UINT__VOID ()
void	ags_cclosure_marshal_BOOLEAN__VOID ()
void	ags_cclosure_marshal_BOOLEAN__UINT_POINTER ()
void	ags_cclosure_marshal_BOOLEAN__POINTER_UINT ()
void	ags_cclosure_marshal_BOOLEAN__OBJECT_OBJECT ()
void	ags_cclosure_marshal_UINT__INT_UINT_POINTER ()
void	ags_cclosure_marshal_DOUBLE__DOUBLE_BOOLEAN ()
void	ags_cclosure_marshal_STRING__STRING_STRING ()
void	ags_cclosure_marshal_STRING__POINTER_STRING_STRING_STRING_STRING_UIN ()
void	ags_cclosure_marshal_POINTER__INT ()
void	ags_cclosure_marshal_POINTER__UINT ()
void	ags_cclosure_marshal_POINTER__POINTER ()
void	ags_cclosure_marshal_POINTER__INT_UINT ()
void	ags_cclosure_marshal_POINTER__UINT_UINT ()
void	ags_cclosure_marshal_POINTER__UINT_POINTER ()
void	ags_cclosure_marshal_POINTER__POINTER_POINTER_POINTER ()
void	ags_cclosure_marshal_POINTER__POINTER_UINT ()
void	ags_cclosure_marshal_POINTER__STRING_STRING ()
void	ags_cclosure_marshal_POINTER__STRING_STRING_STRING ()
void	ags_cclosure_marshal_POINTER__STRING_STRING_STRING_STRING ()
void	ags_cclosure_marshal_POINTER__POINTER_POINTER_UINT ()
void	ags_cclosure_marshal_POINTER__POINTER_UINT_POINTER_POINTER ()
void	ags_cclosure_marshal_POINTER__OBJECT ()
void	ags_cclosure_marshal_POINTER__OBJECT_POINTER_UINT ()
void	ags_cclosure_marshal_POINTER__OBJECT_POINTER_INT64 ()
void	ags_cclosure_marshal_POINTER__OBJECT_OBJECT ()
void	ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_POINTER ()
void	ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_STRING_POI ()
void	ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_UINT_POINT ()

void	ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_STRING_UIN ()
void	ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT_STRING_STRING_ST ()
void	ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT_POINTER_STRING_ ()
void	ags_cclosure_marshal_POINTER__OBJECT_POINTER_POINTER_OBJECT_STRING_ ()
void	ags_cclosure_marshal_POINTER__POINTER_STRING_STRING ()
void	ags_cclosure_marshal_OBJECT__VOID ()
void	ags_cclosure_marshal_OBJECT__UINT_POINTER ()
void	ags_cclosure_marshal_OBJECT__STRING_STRING_STRING ()
void	ags_cclosure_marshal_OBJECT__OBJECT ()
void	ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER ()
void	ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER_POINTER ()
void	ags_cclosure_marshal_OBJECT__OBJECT_OBJECT ()

Description

Functions

ags_cclosure_marshal_VOID__UINT64 ()

```
void
ags_cclosure_marshal_VOID__UINT64 (GClosure *closure,
                                   GValue *return_value,
                                   guint n_param_values,
                                   const GValue *param_values,
                                   gpointer invocation_hint,
                                   gpointer marshal_data);
```

ags_cclosure_marshal_VOID__BOOLEAN_POINTER ()

```
void
ags_cclosure_marshal_VOID__BOOLEAN_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__INT_UINT ()

```
void
ags_cclosure_marshal_VOID__INT_UINT (GClosure *closure,
                                     GValue *return_value,
                                     guint n_param_values,
                                     const GValue *param_values,
                                     gpointer invocation_hint,
                                     gpointer marshal_data);
```

ags_cclosure_marshal_VOID__INT_POINTER ()

```
void
ags_cclosure_marshal_VOID__INT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__INT_UINT_POINTER ()

```
void
ags_cclosure_marshal_VOID__INT_UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_BOOLEAN ()

```
void
ags_cclosure_marshal_VOID__UINT_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT (GClosure *closure,
                                      GValue *return_value,
                                      guint n_param_values,
                                      const GValue *param_values,
                                      gpointer invocation_hint,
                                      gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT_BOOLEAN ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_DOUBLE_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_DOUBLE_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_INT ()

```
void
ags_cclosure_marshal_VOID__UINT_INT (GClosure *closure,
                                     GValue *return_value,
                                     guint n_param_values,
                                     const GValue *param_values,
                                     gpointer invocation_hint,
                                     gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_DOUBLE ()

```
void
ags_cclosure_marshal_VOID__UINT_DOUBLE
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_STRING_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_STRING_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__UINT_POINTER_UINT ()

```
void
ags_cclosure_marshal_VOID__UINT_POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__INT64_UINT ()

```
void
ags_cclosure_marshal_VOID__INT64_UINT (GClosure *closure,
                                       GValue *return_value,
                                       guint n_param_values,
                                       const GValue *param_values,
                                       gpointer invocation_hint,
                                       gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_UINT ()

```
void
ags_cclosure_marshal_VOID__POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_UINT_BOOLEAN ()

```
void
ags_cclosure_marshal_VOID__POINTER_UINT_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__DOUBLE_DOUBLE ()

```
void
ags_cclosure_marshal_VOID__DOUBLE_DOUBLE
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__POINTER_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__STRING_UINT ()

```
void
ags_cclosure_marshal_VOID__STRING_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__STRING_UINT_DOUBLE ()

```
void
ags_cclosure_marshal_VOID__STRING_UINT_DOUBLE
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__STRING_UINT_POINTER ()

```
void
ags_cclosure_marshal_VOID__STRING_UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__STRING_UINT_STRING_DOUBLE ()

```
void
ags_cclosure_marshal_VOID__STRING_UINT_STRING_DOUBLE
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__STRING_POINTER ()

```
void
ags_cclosure_marshal_VOID__STRING_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_VOID__STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_INT ()

```
void
ags_cclosure_marshal_VOID__POINTER_INT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_INT_POINTER ()

```
void
ags_cclosure_marshal_VOID__POINTER_INT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_INT_INT_BOOLEAN ()

```
void
ags_cclosure_marshal_VOID__POINTER_INT_INT_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_POINTER ()

```
void
ags_cclosure_marshal_VOID__POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_UINT_POINTER ()

```
void
ags_cclosure_marshal_VOID__POINTER_UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_POINTER_UINT ()

```
void
ags_cclosure_marshal_VOID__POINTER_POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_POINTER_POINTER ()

```
void
ags_cclosure_marshal_VOID__POINTER_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_STRING_STRING_STRING_INT ()

```
void
ags_cclosure_marshal_VOID__POINTER_STRING_STRING_STRING_INT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__POINTER_OBJECT_OBJECT_STRING_STRING_STRING_UINT_UINT_UINT

```
void
ags_cclosure_marshal_VOID__POINTER_OBJECT_OBJECT_STRING_STRING_STRING_UINT_UINT_UINT_UINT_INT_UINT_U

                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__OBJECT_UINT ()

```
void
ags_cclosure_marshal_VOID__OBJECT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__OBJECT_UINT_UINT_UINT ()

```
void
ags_cclosure_marshal_VOID__OBJECT_UINT_UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__OBJECT_OBJECT ()

```
void
ags_cclosure_marshal_VOID__OBJECT_OBJECT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_INT_BOOLEAN ()

```
void
ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_INT_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_UINT_INT_BOOL

```
void
ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_UINT_INT_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJEC

```
void
ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT_OBJECT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__VOID ()

```
void
ags_cclosure_marshal_POINTER__VOID (GClosure *closure,
                                    GValue *return_value,
                                    guint n_param_values,
                                    const GValue *param_values,
                                    gpointer invocation_hint,
                                    gpointer marshal_data);
```

ags_cclosure_marshal_INT__VOID ()

```
void
ags_cclosure_marshal_INT__VOID (GClosure *closure,
                                GValue *return_value,
                                guint n_param_values,
                                const GValue *param_values,
                                gpointer invocation_hint,
                                gpointer marshal_data);
```

ags_cclosure_marshal_INT64__OBJECT ()

```
void
ags_cclosure_marshal_INT64__OBJECT (GClosure *closure,
                                    GValue *return_value,
                                    guint n_param_values,
                                    const GValue *param_values,
                                    gpointer invocation_hint,
                                    gpointer marshal_data);
```

ags_cclosure_marshal_UINT__VOID ()

```
void
ags_cclosure_marshal_UINT__VOID (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_BOOLEAN__VOID ()

```
void
ags_cclosure_marshal_BOOLEAN__VOID (GClosure *closure,
                                    GValue *return_value,
                                    guint n_param_values,
                                    const GValue *param_values,
                                    gpointer invocation_hint,
                                    gpointer marshal_data);
```

ags_cclosure_marshal_BOOLEAN__UINT_POINTER ()

```
void
ags_cclosure_marshal_BOOLEAN__UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_BOOLEAN__POINTER_UINT ()

```
void
ags_cclosure_marshal_BOOLEAN__POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_BOOLEAN__OBJECT_OBJECT ()

```
void
ags_cclosure_marshal_BOOLEAN__OBJECT_OBJECT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_UINT__INT_UINT_POINTER ()

```
void
ags_cclosure_marshal_UINT__INT_UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_DOUBLE__DOUBLE_BOOLEAN ()

```
void
ags_cclosure_marshal_DOUBLE__DOUBLE_BOOLEAN
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_STRING__STRING_STRING ()

```
void
ags_cclosure_marshal_STRING__STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_STRING__POINTER_STRING_STRING_STRING_STRING_UINT_POINTER ()

```
void
ags_cclosure_marshal_STRING__POINTER_STRING_STRING_STRING_STRING_UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__INT ()

```
void
ags_cclosure_marshal_POINTER__INT (GClosure *closure,
                                   GValue *return_value,
                                   guint n_param_values,
                                   const GValue *param_values,
                                   gpointer invocation_hint,
                                   gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__UINT ()

```
void
ags_cclosure_marshal_POINTER__UINT (GClosure *closure,
                                    GValue *return_value,
                                    guint n_param_values,
                                    const GValue *param_values,
                                    gpointer invocation_hint,
                                    gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__POINTER ()

```
void
ags_cclosure_marshal_POINTER__POINTER (GClosure *closure,
                                       GValue *return_value,
                                       guint n_param_values,
                                       const GValue *param_values,
                                       gpointer invocation_hint,
                                       gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__INT_UINT ()

```
void
ags_cclosure_marshal_POINTER__INT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__UINT_UINT ()

```
void
ags_cclosure_marshal_POINTER__UINT_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__UINT_POINTER ()

```
void
ags_cclosure_marshal_POINTER__UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__POINTER_POINTER_POINTER ()

```
void
ags_cclosure_marshal_POINTER__POINTER_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__POINTER_UINT ()

```
void
ags_cclosure_marshal_POINTER__POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__STRING_STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__STRING_STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__POINTER_POINTER_UINT ()

```
void
ags_cclosure_marshal_POINTER__POINTER_POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__POINTER_UINT_POINTER_POINTER ()

```
void
ags_cclosure_marshal_POINTER__POINTER_UINT_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT ()

```
void
ags_cclosure_marshal_POINTER__OBJECT (GClosure *closure,
                                      GValue *return_value,
                                      guint n_param_values,
                                      const GValue *param_values,
                                      gpointer invocation_hint,
                                      gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_POINTER_UINT ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_POINTER_UINT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_POINTER_INT64 ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_POINTER_INT64
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_OBJECT ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_OBJECT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_POINTER ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_STRING_POINTER ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_STRING_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_UINT_POINTER_POINTER ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_UINT_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_STRING_UINT_POINTER_POINTE

```
void
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_STRING_STRING_UINT_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT_STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT_STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT_POINTER_STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT_POINTER_STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__OBJECT_POINTER_POINTER_OBJECT_STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__OBJECT_POINTER_POINTER_OBJECT_STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_POINTER__POINTER_STRING_STRING ()

```
void
ags_cclosure_marshal_POINTER__POINTER_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__VOID ()

```
void
ags_cclosure_marshal_OBJECT__VOID (GClosure *closure,
                                   GValue *return_value,
                                   guint n_param_values,
                                   const GValue *param_values,
                                   gpointer invocation_hint,
                                   gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__UINT_POINTER ()

```
void
ags_cclosure_marshal_OBJECT__UINT_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__STRING_STRING_STRING ()

```
void
ags_cclosure_marshal_OBJECT__STRING_STRING_STRING
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__OBJECT ()

```
void
ags_cclosure_marshal_OBJECT__OBJECT (GClosure *closure,
                                     GValue *return_value,
                                     guint n_param_values,
                                     const GValue *param_values,
                                     gpointer invocation_hint,
                                     gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER ()

```
void
ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER_POINTER ()

```
void
ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER_POINTER
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

ags_cclosure_marshal_OBJECT__OBJECT_OBJECT ()

```
void
ags_cclosure_marshal_OBJECT__OBJECT_OBJECT
                                (GClosure *closure,
                                 GValue *return_value,
                                 guint n_param_values,
                                 const GValue *param_values,
                                 gpointer invocation_hint,
                                 gpointer marshal_data);
```

Chapter 32

AgsMutable

AgsMutable — unique access to recalls

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_mutable_set_muted ()
#define	AGS_IS_MUTABLE()
#define	AGS_IS_MUTABLE_INTERFACE()
#define	AGS_MUTABLE()
#define	AGS_MUTABLE_GET_INTERFACE()
#define	AGS_MUTABLE_INTERFACE()
GType	ags_mutable_get_type ()

Types and Values

#define	AGS_TYPE_MUTABLE
	AgsMutable
struct	AgsMutableInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsMutable
```

Includes

```
#include <ags/object/ags_mutable.h>
```

Description

The AgsMutable interface gives you a unique access to mutable classes Its purpose is to set muted.

Functions

ags_mutable_set_muted ()

```
void
ags_mutable_set_muted (AgsMutable *agsmutable,
                       gboolean muted);
```

Mute a class instance.

Parameters

agsmutable	an AgsMutable	
muted	if TRUE then muted, else playing	

Since: 3.0.0

AGS_IS_MUTABLE()

```
#define AGS_IS_MUTABLE(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_MUTABLE))
```

AGS_IS_MUTABLE_INTERFACE()

```
#define AGS_IS_MUTABLE_INTERFACE(vtable)    (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_MUTABLE))
```

AGS_MUTABLE()

```
#define AGS_MUTABLE(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MUTABLE, AgsMutable))
```

AGS_MUTABLE_GET_INTERFACE()

```
#define AGS_MUTABLE_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_MUTABLE, AgsMutableInterface))
```

AGS_MUTABLE_INTERFACE()

```
#define AGS_MUTABLE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_MUTABLE, AgsMutableInterface))
```

ags_mutable_get_type ()

```
GType
ags_mutable_get_type ();
```

Types and Values

AGS_TYPE_MUTABLE

```
#define AGS_TYPE_MUTABLE                    (ags_mutable_get_type())
```

AgsMutable

```
typedef struct _AgsMutable AgsMutable;
```

struct AgsMutableInterface

```
struct AgsMutableInterface {
  GTypeInterface ginterface;

  void (*set_muted)(AgsMutable *agsmutable, gboolean muted);
};
```

Chapter 33

AgsPlugin

AgsPlugin — interfacing plugins

Stability Level

Stable, unless otherwise indicated

Functions

gchar *	ags_plugin_get_name ()
void	ags_plugin_set_name ()
gchar *	ags_plugin_get_version ()
void	ags_plugin_set_version ()
gchar *	ags_plugin_get_build_id ()
void	ags_plugin_set_build_id ()
gchar *	ags_plugin_get_xml_type ()
void	ags_plugin_set_xml_type ()
GList *	ags_plugin_get_ports ()
void	ags_plugin_set_ports ()
void	ags_plugin_read ()
xmlNode *	ags_plugin_write ()
#define	AGS_IS_PLUGIN()
#define	AGS_IS_PLUGIN_INTERFACE()
#define	AGS_PLUGIN()
#define	AGS_PLUGIN_GET_INTERFACE()
#define	AGS_PLUGIN_INTERFACE()
GType	ags_plugin_get_type ()

Types and Values

#define	AGS_TYPE_PLUGIN
	AgsPlugin
struct	AgsPluginInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsPlugin
```

Known Implementations

AgsPlugin is implemented by AgsFileLink.

Includes

```
#include <ags/object/ags_plugin.h>
```

Description

The AgsPlugin interface gives you a unique access to classes. It can be used by AgsFile and solves some of the serialization.

Functions

ags_plugin_get_name ()

```
gchar~*
ags_plugin_get_name (AgsPlugin *plugin);
```

Retrieve the name of the plugin.

Parameters

plugin	an *AgsPlugin*

Returns

the plugins name

Since: 3.0.0

ags_plugin_set_name ()

```
void
ags_plugin_set_name (AgsPlugin *plugin,
                     gchar *name);
```

Set the name of the plugin.

Parameters

plugin	an *AgsPlugin*	
name	the name of plugin	

Since: 3.0.0

ags_plugin_get_version ()

```
gchar~*
ags_plugin_get_version (AgsPlugin *plugin);
```

Retrieve the version of the plugin.

Parameters

plugin	an *AgsPlugin*	

Returns

the plugins version

Since: 3.0.0

ags_plugin_set_version ()

```
void
ags_plugin_set_version (AgsPlugin *plugin,
                        gchar *version);
```

Set the version of the plugin.

Parameters

plugin	an *AgsPlugin*	
version	the version of plugin	

Since: 3.0.0

ags_plugin_get_build_id ()

```
gchar~*
ags_plugin_get_build_id (AgsPlugin *plugin);
```

Retrieve the build id of the plugin.

Parameters

plugin	an *AgsPlugin*	

Returns

the plugins build id

Since: 3.0.0

ags_plugin_set_build_id ()

```
void
ags_plugin_set_build_id (AgsPlugin *plugin,
                         gchar *build_id);
```

Set the build id of the plugin.

Parameters

plugin	an *AgsPlugin*	
build_id	the build id of plugin	

Since: 3.0.0

ags_plugin_get_xml_type ()

```
gchar~*
ags_plugin_get_xml_type (AgsPlugin *plugin);
```

Retrieve the xml type of the plugin.

Parameters

plugin	an *AgsPlugin*	

Returns

the plugins xml type

Since: 3.0.0

ags_plugin_set_xml_type ()

```
void
ags_plugin_set_xml_type (AgsPlugin *plugin,
                         gchar *xml_type);
```

Set the build id of the plugin.

Parameters

plugin	an *AgsPlugin*	
xml_type	the build id of plugin	

Since: 3.0.0

ags_plugin_get_ports ()

```
GList~*
ags_plugin_get_ports (AgsPlugin *plugin);
```

Retrieve the ports of the plugin.

Parameters

plugin	an *AgsPlugin*	

Returns

the plugins ports.

[element-type GObject][transfer full]

Since: 3.0.0

ags_plugin_set_ports ()

```
void
ags_plugin_set_ports (AgsPlugin *plugin,
                      GList *ports);
```

Set the build id of the plugin.

Parameters

plugin	an *AgsPlugin*	
ports	the build id of plugin.	*[element-type GObject][transfer none]*

Since: 3.0.0

ags_plugin_read ()

```
void
ags_plugin_read (GObject *file,
                 xmlNode *node,
                 AgsPlugin *plugin);
```

Read of file.

Parameters

file	the GObject	
node	the node	
plugin	the AgsPlugin	

Since: 3.0.0

ags_plugin_write ()

```
xmlNode~*
ags_plugin_write (GObject *file,
                  xmlNode *parent,
                  AgsPlugin *plugin);
```

Write to file.

Parameters

file	the GObject	
parent	the parent node	
plugin	the AgsPlugin	

Returns

the new node you created.

[transfer none]

Since: 3.0.0

AGS_IS_PLUGIN()

```
#define AGS_IS_PLUGIN(obj)                  (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_PLUGIN))
```

AGS_IS_PLUGIN_INTERFACE()

```
#define AGS_IS_PLUGIN_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_PLUGIN))
```

AGS_PLUGIN()

```
#define AGS_PLUGIN(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PLUGIN, AgsPlugin))
```

AGS_PLUGIN_GET_INTERFACE()

```
#define AGS_PLUGIN_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_PLUGIN, AgsPluginInterface))
```

AGS_PLUGIN_INTERFACE()

```
#define AGS_PLUGIN_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_PLUGIN, AgsPluginInterface))
```

ags_plugin_get_type ()

```
GType
ags_plugin_get_type ();
```

Types and Values

AGS_TYPE_PLUGIN

```
#define AGS_TYPE_PLUGIN                     (ags_plugin_get_type())
```

AgsPlugin

```
typedef struct _AgsPlugin AgsPlugin;
```

struct AgsPluginInterface

```
struct AgsPluginInterface {
  GTypeInterface ginterface;

  gchar* (*get_name)(AgsPlugin *plugin);
  void (*set_name)(AgsPlugin *plugin, gchar *name);

  gchar* (*get_version)(AgsPlugin *plugin);
  void (*set_version)(AgsPlugin *plugin, gchar *version);

  gchar* (*get_build_id)(AgsPlugin *plugin);
  void (*set_build_id)(AgsPlugin *plugin, gchar *build_id);

  gchar* (*get_xml_type)(AgsPlugin *plugin);
  void (*set_xml_type)(AgsPlugin *plugin, gchar *xml_type);

  GList* (*get_ports)(AgsPlugin *plugin);
  void (*set_ports)(AgsPlugin *plugin, GList *ports);

  void (*read)(GObject *file,
         xmlNode *node,
         AgsPlugin *plugin);
  xmlNode* (*write)(GObject *file,
        xmlNode *parent,
        AgsPlugin *plugin);
};
```

Chapter 34

AgsPortlet

AgsPortlet — interfacing portlets

Stability Level

Stable, unless otherwise indicated

Functions

GObject *	ags_portlet_get_port ()
void	ags_portlet_set_port ()
GList *	ags_portlet_list_safe_properties ()
void	ags_portlet_safe_get_property ()
void	ags_portlet_safe_set_property ()
#define	AGS_IS_PORTLET()
#define	AGS_IS_PORTLET_INTERFACE()
#define	AGS_PORTLET()
#define	AGS_PORTLET_GET_INTERFACE()
#define	AGS_PORTLET_INTERFACE()
GType	ags_portlet_get_type ()

Types and Values

#define	AGS_TYPE_PORTLET
	AgsPortlet
struct	AgsPortletInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsPortlet
```

Includes

```
#include <ags/object/ags_portlet.h>
```

Description

The AgsPortlet interface is intended to use with ports containing GObject. Whereas its should implement this interface.

Functions

ags_portlet_get_port ()

```
GObject~*
ags_portlet_get_port (AgsPortlet *portlet);
```

Get port.

Parameters

portlet	the AgsPortlet

Returns

the GObject.

[transfer full]

Since: 3.0.0

ags_portlet_set_port ()

```
void
ags_portlet_set_port (AgsPortlet *portlet,
                      GObject *port);
```

Set port.

Parameters

portlet	the AgsPortlet
port	the GObject

Since: 3.0.0

ags_portlet_list_safe_properties ()

```
GList~*
ags_portlet_list_safe_properties (AgsPortlet *portlet);
```

Retrieve thread-safe properties.

Parameters

portlet	the AgsPortlet	

Returns

a GList containing properties.

[element-type utf8][transfer full]

Since: 3.0.0

ags_portlet_safe_get_property ()

```
void
ags_portlet_safe_get_property (AgsPortlet *portlet,
                               gchar *property_name,
                               GValue *value);
```

Get property thread safe.

Parameters

portlet	the AgsPortlet	
property_name	propertie's name	
value	the GValue	

Since: 3.0.0

ags_portlet_safe_set_property ()

```
void
ags_portlet_safe_set_property (AgsPortlet *portlet,
                               gchar *property_name,
                               GValue *value);
```

Set property thread safe.

Parameters

portlet	the AgsPortlet	
property_name	propertie's name	
value	the GValue	

Since: 3.0.0

AGS_IS_PORTLET()

```
#define AGS_IS_PORTLET(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_PORTLET))
```

AGS_IS_PORTLET_INTERFACE()

```
#define AGS_IS_PORTLET_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_PORTLET))
```

AGS_PORTLET()

```
#define AGS_PORTLET(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PORTLET, AgsPortlet))
```

AGS_PORTLET_GET_INTERFACE()

```
#define AGS_PORTLET_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_PORTLET, AgsPortletInterface))
```

AGS_PORTLET_INTERFACE()

```
#define AGS_PORTLET_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_PORTLET, AgsPortletInterface))
```

ags_portlet_get_type ()

```
GType
ags_portlet_get_type ();
```

Types and Values

AGS_TYPE_PORTLET

```
#define AGS_TYPE_PORTLET                     (ags_portlet_get_type())
```

AgsPortlet

```
typedef struct _AgsPortlet AgsPortlet;
```

struct AgsPortletInterface

```
struct AgsPortletInterface {
  GTypeInterface ginterface;

  GObject* (*get_port)(AgsPortlet *portlet);
  void (*set_port)(AgsPortlet *portlet, GObject *port);

  GList* (*list_safe_properties)(AgsPortlet *portlet);

  void (*safe_get_property)(AgsPortlet *portlet, gchar *property_name, GValue *value);
  void (*safe_set_property)(AgsPortlet *portlet, gchar *property_name, GValue *value);
};
```

Chapter 35

AgsSeekable

AgsSeekable — unique access to seekable classes

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_seekable_seek ()
#define	AGS_IS_SEEKABLE()
#define	AGS_IS_SEEKABLE_INTERFACE()
#define	AGS_SEEKABLE()
#define	AGS_SEEKABLE_GET_INTERFACE()
#define	AGS_SEEKABLE_INTERFACE()
GType	ags_seek_type_get_type ()
GType	ags_seekable_get_type ()

Signals

void	seek	Run Last

Types and Values

enum	AgsSeekType
#define	AGS_TYPE_SEEKABLE
#define	AGS_TYPE_SEEK_TYPE
	AgsSeekable
struct	AgsSeekableInterface

Object Hierarchy

```
    GEnum
    &#x2570;&#x2500;&#x2500; AgsSeekType
    GInterface
    &#x2570;&#x2500;&#x2500; AgsSeekable
```

Includes

```
#include <ags/object/ags_seekable.h>
```

Description

The AgsSeekable interface gives you the AgsSeekable::seek() signal what notifies about changed offset of pattern or notation.

Functions

ags_seekable_seek ()

```
void
ags_seekable_seek (AgsSeekable *seekable,
                   gint64 offset,
                   guint whence);
```

Seek.

Parameters

seekable	the AgsSeekable interface	
offset	the offset	
whence	the direction, see AgsSeekType	

Since: 3.0.0

AGS_IS_SEEKABLE()

```
#define AGS_IS_SEEKABLE(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SEEKABLE))
```

AGS_IS_SEEKABLE_INTERFACE()

```
#define AGS_IS_SEEKABLE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SEEKABLE))
```

AGS_SEEKABLE()

```
#define AGS_SEEKABLE(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SEEKABLE, AgsSeekable))
```

AGS_SEEKABLE_GET_INTERFACE()

```
#define AGS_SEEKABLE_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SEEKABLE, AgsSeekableInterface))
```

AGS_SEEKABLE_INTERFACE()

```
#define AGS_SEEKABLE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SEEKABLE, AgsSeekableInterface))
```

ags_seek_type_get_type ()

```
GType
ags_seek_type_get_type ();
```

ags_seekable_get_type ()

```
GType
ags_seekable_get_type ();
```

Types and Values

enum AgsSeekType

Seek type.

Members

AGS_SEEK_CUR	seek from current position

AGS_SEEK_SET	seek by setting absolute position
AGS_SEEK_END	seek from end

AGS_TYPE_SEEKABLE

```
#define AGS_TYPE_SEEKABLE                    (ags_seekable_get_type())
```

AGS_TYPE_SEEK_TYPE

```
#define AGS_TYPE_SEEK_TYPE                   (ags_seek_type_get_type())
```

AgsSeekable

```
typedef struct _AgsSeekable AgsSeekable;
```

struct AgsSeekableInterface

```
struct AgsSeekableInterface {
  GTypeInterface ginterface;

  void (*seek)(AgsSeekable *seekable, gint64 offset, guint whence);
};
```

Signal Details

The "seek" signal

```
void
user_function (AgsSeekable *seekable,
               gint64       offset,
               guint        whence,
               gpointer     user_data)
```

The ::seek signal notifies about changed position of sequencer.

Parameters

seekable	the GObject sub-type implementing AgsSeekable	
offset	the offset	
whence	the direction, see AgsSeekType	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 36

AgsSequencer

AgsSequencer — unique access to sequencers

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_sequencer_set_device ()
gchar *	ags_sequencer_get_device ()
void	ags_sequencer_list_cards ()
gboolean	ags_sequencer_is_starting ()
gboolean	ags_sequencer_is_playing ()
gboolean	ags_sequencer_is_recording ()
void	ags_sequencer_play_init ()
void	ags_sequencer_play ()
void	ags_sequencer_record_init ()
void	ags_sequencer_record ()
void	ags_sequencer_stop ()
void	ags_sequencer_tic ()
void	ags_sequencer_offset_changed ()
void *	ags_sequencer_get_buffer ()
void *	ags_sequencer_get_next_buffer ()
void	ags_sequencer_lock_buffer ()
void	ags_sequencer_unlock_buffer ()
void	ags_sequencer_set_bpm ()
gdouble	ags_sequencer_get_bpm ()
void	ags_sequencer_set_delay_factor ()
gdouble	ags_sequencer_get_delay_factor ()
void	ags_sequencer_set_start_note_offset ()
guint	ags_sequencer_get_start_note_offset ()
void	ags_sequencer_set_note_offset ()
guint	ags_sequencer_get_note_offset ()
#define	AGS_IS_SEQUENCER()
#define	AGS_IS_SEQUENCER_INTERFACE()
#define	AGS_SEQUENCER()

#define	AGS_SEQUENCER_GET_INTERFACE()
#define	AGS_SEQUENCER_INTERFACE()
GType	ags_sequencer_get_type ()

Signals

void	offset-changed	Run Last
void	tic	Run Last

Types and Values

#define	AGS_SEQUENCER_MAX_MIDI_CHANNELS
#define	AGS_SEQUENCER_MAX_MIDI_KEYS
#define	AGS_SEQUENCER_DEFAULT_SAMPLERATE
#define	AGS_SEQUENCER_DEFAULT_BUFFER_SIZE
#define	AGS_SEQUENCER_DEFAULT_DEVICE
#define	AGS_SEQUENCER_DEFAULT_BPM
#define	AGS_SEQUENCER_DEFAULT_DELAY_FACTOR
#define	AGS_SEQUENCER_DEFAULT_JIFFIE
#define	AGS_SEQUENCER_DEFAULT_TACT
#define	AGS_SEQUENCER_DEFAULT_TACT_JIFFIE
#define	AGS_SEQUENCER_DEFAULT_TACTRATE
#define	AGS_SEQUENCER_DEFAULT_SCALE
#define	AGS_SEQUENCER_DEFAULT_DELAY
#define	AGS_SEQUENCER_DEFAULT_PERIOD
#define	AGS_SEQUENCER_DEFAULT_LATENCY
#define	AGS_TYPE_SEQUENCER
	AgsSequencer
struct	AgsSequencerInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsSequencer
```

Includes

```
#include <ags/object/ags_sequencer.h>
```

Description

The AgsSequencer interface gives you a unique access to MIDI devices.

Functions

ags_sequencer_set_device ()

```
void
ags_sequencer_set_device (AgsSequencer *sequencer,
                          gchar *card_id);
```

Set device.

Parameters

sequencer	the AgsSequencer	
card_id	the device to set	

Since: 3.0.0

ags_sequencer_get_device ()

```
gchar~*
ags_sequencer_get_device (AgsSequencer *sequencer);
```

Get device.

Parameters

sequencer	the AgsSequencer	

Returns

the device's identifier

Since: 3.0.0

ags_sequencer_list_cards ()

```
void
ags_sequencer_list_cards (AgsSequencer *sequencer,
                          GList **card_id,
                          GList **card_name);
```

Retrieve *card_id* and *card_name* as a list of strings.

Parameters

sequencer	the AgsSequencer	
card_id	a list containing card ids.	*[element-type utf8][out callee-allocates][array zero-terminated=1][transfer full]*
card_name	a list containing card names.	*[element-type utf8][out callee-allocates][array zero-terminated=1][transfer full]*

Since: 3.0.0

ags_sequencer_is_starting ()

```
gboolean
ags_sequencer_is_starting (AgsSequencer *sequencer);
```

Get starting.

Parameters

sequencer	the AgsSequencer	

Returns

TRUE if starting, else FALSE

Since: 3.0.0

ags_sequencer_is_playing ()

```
gboolean
ags_sequencer_is_playing (AgsSequencer *sequencer);
```

Get playing.

Parameters

sequencer	the AgsSequencer	

Returns

TRUE if playing, else FALSE

Since: 3.0.0

ags_sequencer_is_recording ()

```
gboolean
ags_sequencer_is_recording (AgsSequencer *sequencer);
```

Get recording.

Parameters

sequencer	the AgsSequencer	

Returns

TRUE if recording, else FALSE

Since: 3.0.0

ags_sequencer_play_init ()

```
void
ags_sequencer_play_init (AgsSequencer *sequencer,
                         GError **error);
```

Initializes the sequencer for playback.

Parameters

sequencer	the AgsSequencer	
error	an error that may occure	

Since: 3.0.0

ags_sequencer_play ()

```
void
ags_sequencer_play (AgsSequencer *sequencer,
                    GError **error);
```

Plays the current buffer of sequencer.

Parameters

sequencer	the AgsSequencer	
error	an error that may occure	

Since: 3.0.0

ags_sequencer_record_init ()

```
void
ags_sequencer_record_init (AgsSequencer *sequencer,
                           GError **error);
```

Initializes the sequencer for recording.

Parameters

sequencer	the AgsSequencer	
error	an error that may occure	

Since: 3.0.0

ags_sequencer_record ()

```
void
ags_sequencer_record (AgsSequencer *sequencer,
                      GError **error);
```

Records the current buffer of sequencer.

Parameters

sequencer	the AgsSequencer	
error	an error that may occure	

Since: 3.0.0

ags_sequencer_stop ()

```
void
ags_sequencer_stop (AgsSequencer *sequencer);
```

Stops the sequencer from playing to it.

Parameters

sequencer	the AgsSequencer	

Since: 3.0.0

ags_sequencer_tic ()

```
void
ags_sequencer_tic (AgsSequencer *sequencer);
```

Every call to play may generate a tic.

Parameters

sequencer	the AgsSequencer	

Since: 3.0.0

ags_sequencer_offset_changed ()

```
void
ags_sequencer_offset_changed (AgsSequencer *sequencer,
                              guint note_offset);
```

Callback when counter expires minor note offset.

Parameters

sequencer	the AgsSequencer	
note_offset	the note offset	

Since: 3.0.0

ags_sequencer_get_buffer ()

```
void~*
ags_sequencer_get_buffer (AgsSequencer *sequencer,
                          guint *buffer_length);
```

Get current playback buffer.

Parameters

sequencer	the AgsSequencer	
buffer_length	the buffer's length	

Returns

current playback buffer

Since: 3.0.0

ags_sequencer_get_next_buffer ()

```
void~*
ags_sequencer_get_next_buffer (AgsSequencer *sequencer,
                               guint *buffer_length);
```

Get future playback buffer.

Parameters

sequencer	the AgsSequencer	
buffer_length	the buffer's length	

Returns

next playback buffer

Since: 3.0.0

ags_sequencer_lock_buffer ()

```
void
ags_sequencer_lock_buffer (AgsSequencer *sequencer,
                           void *buffer);
```

Lock *buffer* .

Parameters

sequencer	the AgsSequencer	
buffer	the buffer to lock	

Since: 3.0.0

ags_sequencer_unlock_buffer ()

```
void
ags_sequencer_unlock_buffer (AgsSequencer *sequencer,
                             void *buffer);
```

Unlock *buffer* .

Parameters

sequencer	the AgsSequencer	
buffer	the buffer to unlock	

Since: 3.0.0

ags_sequencer_set_bpm ()

```
void
ags_sequencer_set_bpm (AgsSequencer *sequencer,
                       gdouble bpm);
```

Set current playback bpm.

Parameters

sequencer	the AgsSequencer	
bpm	the bpm to set	

Since: 3.0.0

ags_sequencer_get_bpm ()

```
gdouble
ags_sequencer_get_bpm (AgsSequencer *sequencer);
```

Get current playback bpm.

Parameters

sequencer	the AgsSequencer

Returns

bpm

Since: 3.0.0

ags_sequencer_set_delay_factor ()

```
void
ags_sequencer_set_delay_factor (AgsSequencer *sequencer,
                                gdouble delay_factor);
```

Set current playback delay factor.

Parameters

sequencer	the AgsSequencer
delay_factor	the delay factor to set

Since: 3.0.0

ags_sequencer_get_delay_factor ()

```
gdouble
ags_sequencer_get_delay_factor (AgsSequencer *sequencer);
```

Get current playback delay factor.

Parameters

sequencer	the AgsSequencer

Returns

delay factor

Since: 3.0.0

ags_sequencer_set_start_note_offset ()

```
void
ags_sequencer_set_start_note_offset (AgsSequencer *sequencer,
                                     guint start_note_offset);
```

Set start playback note offset.

Parameters

sequencer	the AgsSequencer	
start_note_offset	the start note offset to set	

Since: 3.0.0

ags_sequencer_get_start_note_offset ()

```
guint
ags_sequencer_get_start_note_offset (AgsSequencer *sequencer);
```

Get start playback note offset.

Parameters

sequencer	the AgsSequencer	

Returns

the start note offset

Since: 3.0.0

ags_sequencer_set_note_offset ()

```
void
ags_sequencer_set_note_offset (AgsSequencer *sequencer,
                               guint note_offset);
```

Set current playback note offset.

Parameters

sequencer	the AgsSequencer	
note_offset	the note offset to set	

Since: 3.0.0

ags_sequencer_get_note_offset ()

```
guint
ags_sequencer_get_note_offset (AgsSequencer *sequencer);
```

Get current playback note offset.

Parameters

sequencer	the AgsSequencer

Returns

offset

Since: 3.0.0

AGS_IS_SEQUENCER()

```
#define AGS_IS_SEQUENCER(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SEQUENCER))
```

AGS_IS_SEQUENCER_INTERFACE()

```
#define AGS_IS_SEQUENCER_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SEQUENCER))
```

AGS_SEQUENCER()

```
#define AGS_SEQUENCER(obj)                       (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SEQUENCER, AgsSequencer))
```

AGS_SEQUENCER_GET_INTERFACE()

```
#define AGS_SEQUENCER_GET_INTERFACE(obj)         (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SEQUENCER, AgsSequencerInterface))
```

AGS_SEQUENCER_INTERFACE()

```
#define AGS_SEQUENCER_INTERFACE(vtable)          (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SEQUENCER, AgsSequencerInterface))
```

ags_sequencer_get_type ()

```
GType
ags_sequencer_get_type ();
```

Types and Values

AGS_SEQUENCER_MAX_MIDI_CHANNELS

```
#define AGS_SEQUENCER_MAX_MIDI_CHANNELS (16)
```

AGS_SEQUENCER_MAX_MIDI_KEYS

```
#define AGS_SEQUENCER_MAX_MIDI_KEYS (128)
```

AGS_SEQUENCER_DEFAULT_SAMPLERATE

```
#define AGS_SEQUENCER_DEFAULT_SAMPLERATE (44100.0)
```

AGS_SEQUENCER_DEFAULT_BUFFER_SIZE

```
#define AGS_SEQUENCER_DEFAULT_BUFFER_SIZE (944)
```

AGS_SEQUENCER_DEFAULT_DEVICE

```
#define AGS_SEQUENCER_DEFAULT_DEVICE "hw:0"
```

AGS_SEQUENCER_DEFAULT_BPM

```
#define AGS_SEQUENCER_DEFAULT_BPM (120.0)
```

AGS_SEQUENCER_DEFAULT_DELAY_FACTOR

```
#define AGS_SEQUENCER_DEFAULT_DELAY_FACTOR (1.0 / 4.0)
```

AGS_SEQUENCER_DEFAULT_JIFFIE

```
#define AGS_SEQUENCER_DEFAULT_JIFFIE ((double) AGS_SEQUENCER_DEFAULT_SAMPLERATE / (double)  ↩
    AGS_SEQUENCER_DEFAULT_BUFFER_SIZE)
```

AGS_SEQUENCER_DEFAULT_TACT

```
#define AGS_SEQUENCER_DEFAULT_TACT (1.0 / 1.0)
```

AGS_SEQUENCER_DEFAULT_TACT_JIFFIE

```
#define AGS_SEQUENCER_DEFAULT_TACT_JIFFIE (60.0 / AGS_SEQUENCER_DEFAULT_BPM *  ↩
    AGS_SEQUENCER_DEFAULT_TACT)
```

AGS_SEQUENCER_DEFAULT_TACTRATE

```
#define AGS_SEQUENCER_DEFAULT_TACTRATE (1.0 / AGS_SEQUENCER_DEFAULT_TACT_JIFFIE)
```

AGS_SEQUENCER_DEFAULT_SCALE

```
#define AGS_SEQUENCER_DEFAULT_SCALE (1.0)
```

AGS_SEQUENCER_DEFAULT_DELAY

```
#define AGS_SEQUENCER_DEFAULT_DELAY (AGS_SEQUENCER_DEFAULT_JIFFIE * (60.0 /  ↩
    AGS_SEQUENCER_DEFAULT_BPM))
```

AGS_SEQUENCER_DEFAULT_PERIOD

```
#define AGS_SEQUENCER_DEFAULT_PERIOD (64.0)
```

AGS_SEQUENCER_DEFAULT_LATENCY

```
#define AGS_SEQUENCER_DEFAULT_LATENCY (400)
```

AGS_TYPE_SEQUENCER

```
#define AGS_TYPE_SEQUENCER                    (ags_sequencer_get_type())
```

AgsSequencer

```
typedef struct _AgsSequencer AgsSequencer;
```

struct AgsSequencerInterface

```
struct AgsSequencerInterface {
  GTypeInterface ginterface;

  void (*set_device)(AgsSequencer *sequencer,
         gchar *card_id);
  gchar* (*get_device)(AgsSequencer *sequencer);

  void (*list_cards)(AgsSequencer *sequencer,
         GList **card_id, GList **card_name);

  gboolean (*is_starting)(AgsSequencer *sequencer);
  gboolean (*is_playing)(AgsSequencer *sequencer);
  gboolean (*is_recording)(AgsSequencer *sequencer);

  void (*play_init)(AgsSequencer *sequencer,
        GError **error);
  void (*play)(AgsSequencer *sequencer,
         GError **error);

  void (*record_init)(AgsSequencer *sequencer,
          GError **error);
  void (*record)(AgsSequencer *sequencer,
```

```
    GError **error);

  void (*stop)(AgsSequencer *sequencer);

  void (*tic)(AgsSequencer *sequencer);
  void (*offset_changed)(AgsSequencer *sequencer,
       guint note_offset);

  void* (*get_buffer)(AgsSequencer *sequencer,
         guint *buffer_length);
  void* (*get_next_buffer)(AgsSequencer *sequencer,
        guint *buffer_length);

  void (*lock_buffer)(AgsSequencer *sequencer,
         void *buffer);
  void (*unlock_buffer)(AgsSequencer *sequencer,
      void *buffer);

  void (*set_bpm)(AgsSequencer *sequencer,
      gdouble bpm);
  gdouble (*get_bpm)(AgsSequencer *sequencer);

  void (*set_delay_factor)(AgsSequencer *sequencer,
        gdouble delay_factor);
  gdouble (*get_delay_factor)(AgsSequencer *sequencer);

  void (*set_start_note_offset)(AgsSequencer *sequencer,
       guint start_note_offset);
  guint (*get_start_note_offset)(AgsSequencer *sequencer);

  void (*set_note_offset)(AgsSequencer *sequencer,
       guint note_offset);
  guint (*get_note_offset)(AgsSequencer *sequencer);
};
```

Signal Details

The "offset-changed" signal

```
void
user_function (AgsSequencer *sequencer,
               guint         note_offset,
               gpointer      user_data)
```

The ::offset-changed signal notifies about changed position within notation.

Parameters

sequencer	the AgsSequencer object	
note_offset	new notation offset	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "tic" signal

```
void
user_function (AgsSequencer *sequencer,
               gpointer      user_data)
```

The ::tic signal is emitted every tic of the sequencer. This notifies about a newly played buffer.

Parameters

sequencer	the AgsSequencer object	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 37

AgsSoundcard

AgsSoundcard — unique access to soundcards

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_soundcard_set_device ()
gchar *	ags_soundcard_get_device ()
void	ags_soundcard_set_presets ()
void	ags_soundcard_get_presets ()
void	ags_soundcard_list_cards ()
void	ags_soundcard_pcm_info ()
guint	ags_soundcard_get_capability ()
gboolean	ags_soundcard_is_available ()
gboolean	ags_soundcard_is_starting ()
gboolean	ags_soundcard_is_playing ()
gboolean	ags_soundcard_is_recording ()
gchar *	ags_soundcard_get_uptime ()
void	ags_soundcard_play_init ()
void	ags_soundcard_play ()
void	ags_soundcard_record_init ()
void	ags_soundcard_record ()
void	ags_soundcard_stop ()
void	ags_soundcard_tic ()
void	ags_soundcard_offset_changed ()
void *	ags_soundcard_get_buffer ()
void *	ags_soundcard_get_next_buffer ()
void *	ags_soundcard_get_prev_buffer ()
void	ags_soundcard_lock_buffer ()
void	ags_soundcard_unlock_buffer ()
void	ags_soundcard_set_bpm ()
gdouble	ags_soundcard_get_bpm ()
void	ags_soundcard_set_delay_factor ()
gdouble	ags_soundcard_get_delay_factor ()

gdouble	ags_soundcard_get_absolute_delay ()
gdouble	ags_soundcard_get_delay ()
guint	ags_soundcard_get_attack ()
guint	ags_soundcard_get_delay_counter ()
void	ags_soundcard_set_start_note_offset ()
guint	ags_soundcard_get_start_note_offset ()
void	ags_soundcard_set_note_offset ()
guint	ags_soundcard_get_note_offset ()
void	ags_soundcard_set_note_offset_absolute ()
guint	ags_soundcard_get_note_offset_absolute ()
void	ags_soundcard_set_loop ()
void	ags_soundcard_get_loop ()
guint	ags_soundcard_get_loop_offset ()
guint	ags_soundcard_get_sub_block_count ()
gboolean	ags_soundcard_trylock_sub_block ()
void	ags_soundcard_unlock_sub_block ()
#define	AGS_IS_SOUNDCARD()
#define	AGS_IS_SOUNDCARD_INTERFACE()
#define	AGS_SOUNDCARD()
#define	AGS_SOUNDCARD_GET_INTERFACE()
#define	AGS_SOUNDCARD_INTERFACE()
GType	ags_soundcard_capability_get_type ()
GType	ags_soundcard_format_get_type ()
GType	ags_soundcard_get_type ()

Signals

void	offset-changed	Run Last
void	stop	Run Last
void	tic	Run Last

Types and Values

#define	AGS_SOUNDCARD_DEFAULT_DSP_CHANNELS
#define	AGS_SOUNDCARD_MIN_DSP_CHANNELS
#define	AGS_SOUNDCARD_MAX_DSP_CHANNELS
#define	AGS_SOUNDCARD_DEFAULT_PCM_CHANNELS
#define	AGS_SOUNDCARD_MIN_PCM_CHANNELS
#define	AGS_SOUNDCARD_MAX_PCM_CHANNELS
#define	AGS_SOUNDCARD_DEFAULT_SAMPLERATE
#define	AGS_SOUNDCARD_MIN_SAMPLERATE
#define	AGS_SOUNDCARD_MAX_SAMPLERATE
#define	AGS_SOUNDCARD_DEFAULT_FORMAT
#define	AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE
#define	AGS_SOUNDCARD_MIN_BUFFER_SIZE
#define	AGS_SOUNDCARD_MAX_BUFFER_SIZE
#define	AGS_SOUNDCARD_DEFAULT_BPM
#define	AGS_SOUNDCARD_DEFAULT_DELAY_FACTOR
#define	AGS_SOUNDCARD_DEFAULT_JIFFIE
#define	AGS_SOUNDCARD_DEFAULT_TACT
#define	AGS_SOUNDCARD_DEFAULT_TACT_JIFFIE
#define	AGS_SOUNDCARD_DEFAULT_TACTRATE

#define	AGS_SOUNDCARD_DEFAULT_SCALE
#define	AGS_SOUNDCARD_DEFAULT_DELAY
#define	AGS_SOUNDCARD_DEFAULT_PERIOD
#define	AGS_SOUNDCARD_DEFAULT_LOOP_LEFT
#define	AGS_SOUNDCARD_DEFAULT_LOOP_RIGHT
#define	AGS_SOUNDCARD_DEFAULT_OVERCLOCK
#define	AGS_SOUNDCARD_DEFAULT_SUB_BLOCK_COUNT
enum	AgsSoundcardFormat
enum	AgsSoundcardCapability
#define	AGS_TYPE_SOUNDCARD
#define	AGS_TYPE_SOUNDCARD_CAPABILITY
#define	AGS_TYPE_SOUNDCARD_FORMAT
	AgsSoundcard
struct	AgsSoundcardInterface

Object Hierarchy

```
    GEnum
    &#x2570;&#x2500;&#x2500; AgsSoundcardFormat
    GFlags
    &#x2570;&#x2500;&#x2500; AgsSoundcardCapability
    GInterface
    &#x2570;&#x2500;&#x2500; AgsSoundcard
```

Includes

```
#include <ags/object/ags_soundcard.h>
```

Description

The AgsSoundcard interface gives you a unique access to audio devices.

Functions

ags_soundcard_set_device ()

```
void
ags_soundcard_set_device (AgsSoundcard *soundcard,
                          gchar *card_id);
```

Set device.

Parameters

soundcard	the AgsSoundcard	
card_id	the device to set	

Since: 3.0.0

ags_soundcard_get_device ()

```
gchar~*
ags_soundcard_get_device (AgsSoundcard *soundcard);
```

Get device.

Parameters

soundcard	the AgsSoundcard

Returns

the device's identifier

Since: 3.0.0

ags_soundcard_set_presets ()

```
void
ags_soundcard_set_presets (AgsSoundcard *soundcard,
                           guint channels,
                           guint rate,
                           guint buffer_size,
                           guint format);
```

Set presets.

Parameters

soundcard	the AgsSoundcard	
channels	the audio channels	
rate	the samplerate	
buffer_size	the buffer size	
format	the format	

Since: 3.0.0

ags_soundcard_get_presets ()

```
void
ags_soundcard_get_presets (AgsSoundcard *soundcard,
                           guint *channels,
                           guint *rate,
                           guint *buffer_size,
                           guint *format);
```

Get presets.

Parameters

soundcard	the AgsSoundcard	
channels	the audio channels.	*[out]*
rate	the samplerate.	*[out]*
buffer_size	the buffer size.	*[out]*
format	the format.	*[out]*

Since: 3.0.0

ags_soundcard_list_cards ()

```
void
ags_soundcard_list_cards (AgsSoundcard *soundcard,
                          GList **card_id,
                          GList **card_name);
```

Retrieve `card_id` and `card_name` as a list of strings.

Parameters

soundcard	the AgsSoundcard	
card_id	a list containing card ids.	*[element-type utf8][out callee-allocates][array zero-terminated=1][transfer full]*
card_name	a list containing card names.	*[element-type utf8][out callee-allocates][array zero-terminated=1][transfer full]*

Since: 3.0.0

ags_soundcard_pcm_info ()

```
void
ags_soundcard_pcm_info (AgsSoundcard *soundcard,
                        gchar *card_id,
                        guint *channels_min,
                        guint *channels_max,
                        guint *rate_min,
                        guint *rate_max,
                        guint *buffer_size_min,
                        guint *buffer_size_max,
                        GError **error);
```

Retrieve detailed information of `card_id` soundcard.

Parameters

soundcard	the AgsSoundcard	
card_id	the selected soundcard by its string identifier	
channels_min	min channels supported.	*[out]*
channels_max	max channels supported.	*[out]*

rate_min	min samplerate supported.	*[out]*
rate_max	max samplerate supported.	*[out]*
buffer_size_min	min buffer size supported by soundcard.	*[out]*
buffer_size_max	max buffer size supported by soundcard.	*[out]*
error	an error that may occure	

Since: 3.0.0

ags_soundcard_get_capability ()

```
guint
ags_soundcard_get_capability (AgsSoundcard *soundcard);
```

Retrieve AgsSoundcardCapability information.

Parameters

soundcard	the AgsSoundcard

Returns

the capablities flags

Since: 3.0.0

ags_soundcard_is_available ()

```
gboolean
ags_soundcard_is_available (AgsSoundcard *soundcard);
```

Get available.

Parameters

soundcard	the AgsSoundcard

Returns

TRUE if available, else FALSE

Since: 3.0.0

ags_soundcard_is_starting ()

```
gboolean
ags_soundcard_is_starting (AgsSoundcard *soundcard);
```

Get starting.

Parameters

soundcard	the AgsSoundcard

Returns

TRUE if starting, else FALSE

Since: 3.0.0

ags_soundcard_is_playing ()

```
gboolean
ags_soundcard_is_playing (AgsSoundcard *soundcard);
```

Get playing.

Parameters

soundcard	the AgsSoundcard

Returns

TRUE if playing, else FALSE

Since: 3.0.0

ags_soundcard_is_recording ()

```
gboolean
ags_soundcard_is_recording (AgsSoundcard *soundcard);
```

Get recording.

Parameters

soundcard	the AgsSoundcard

Returns

TRUE if recording, else FALSE

Since: 3.0.0

ags_soundcard_get_uptime ()

```
gchar~*
ags_soundcard_get_uptime (AgsSoundcard *soundcard);
```

Get playback time as string.

Parameters

soundcard	the AgsSoundcard	

Returns

playback time as string

Since: 3.0.0

ags_soundcard_play_init ()

```
void
ags_soundcard_play_init (AgsSoundcard *soundcard,
                         GError **error);
```

Initializes the soundcard for playback.

Parameters

soundcard	the AgsSoundcard	
error	an error that may occure	

Since: 3.0.0

ags_soundcard_play ()

```
void
ags_soundcard_play (AgsSoundcard *soundcard,
                    GError **error);
```

Plays the current buffer of soundcard.

Parameters

soundcard	the AgsSoundcard	
error	an error that may occure	

Since: 3.0.0

ags_soundcard_record_init ()

```
void
ags_soundcard_record_init (AgsSoundcard *soundcard,
                           GError **error);
```

Initializes the soundcard for recordback.

Parameters

soundcard	the AgsSoundcard	
error	an error that may occure	

Since: 3.0.0

ags_soundcard_record ()

```
void
ags_soundcard_record (AgsSoundcard *soundcard,
                      GError **error);
```

Records the current buffer of soundcard.

Parameters

soundcard	the AgsSoundcard	
error	an error that may occure	

Since: 3.0.0

ags_soundcard_stop ()

```
void
ags_soundcard_stop (AgsSoundcard *soundcard);
```

Stops the soundcard from playing to it.

Parameters

soundcard	the AgsSoundcard	

Since: 3.0.0

ags_soundcard_tic ()

```
void
ags_soundcard_tic (AgsSoundcard *soundcard);
```

Every call to play may generate a tic.

Parameters

soundcard	the AgsSoundcard	

Since: 3.0.0

ags_soundcard_offset_changed ()

```
void
ags_soundcard_offset_changed (AgsSoundcard *soundcard,
                              guint note_offset);
```

Callback when counter expires minor note offset.

Parameters

soundcard	the AgsSoundcard	
note_offset	the current note offset	

Since: 3.0.0

ags_soundcard_get_buffer ()

```
void~*
ags_soundcard_get_buffer (AgsSoundcard *soundcard);
```

Get current playback buffer.

Parameters

soundcard	the AgsSoundcard	

Returns

current playback buffer

Since: 3.0.0

ags_soundcard_get_next_buffer ()

```
void~*
ags_soundcard_get_next_buffer (AgsSoundcard *soundcard);
```

Get future playback buffer.

Parameters

soundcard	the AgsSoundcard	

Returns

next playback buffer

Since: 3.0.0

ags_soundcard_get_prev_buffer ()

```
void~*
ags_soundcard_get_prev_buffer (AgsSoundcard *soundcard);
```

Get future playback buffer.

Parameters

soundcard	the AgsSoundcard

Returns

prev playback buffer

Since: 3.0.0

ags_soundcard_lock_buffer ()

```
void
ags_soundcard_lock_buffer (AgsSoundcard *soundcard,
                           void *buffer);
```

Lock *buffer* .

Parameters

soundcard	the AgsSoundcard
buffer	the buffer to lock

Since: 3.0.0

ags_soundcard_unlock_buffer ()

```
void
ags_soundcard_unlock_buffer (AgsSoundcard *soundcard,
                             void *buffer);
```

Unlock *buffer* .

Parameters

soundcard	the AgsSoundcard
buffer	the buffer to unlock

Since: 3.0.0

ags_soundcard_set_bpm ()

```
void
ags_soundcard_set_bpm (AgsSoundcard *soundcard,
                       gdouble bpm);
```

Set current playback bpm.

Parameters

soundcard	the AgsSoundcard	
bpm	the bpm to set	

Since: 3.0.0

ags_soundcard_get_bpm ()

```
gdouble
ags_soundcard_get_bpm (AgsSoundcard *soundcard);
```

Get current playback bpm.

Parameters

soundcard	the AgsSoundcard	

Returns

bpm

Since: 3.0.0

ags_soundcard_set_delay_factor ()

```
void
ags_soundcard_set_delay_factor (AgsSoundcard *soundcard,
                                gdouble delay_factor);
```

Set current playback delay factor.

Parameters

soundcard	the AgsSoundcard	
delay_factor	the delay factor to set	

Since: 3.0.0

ags_soundcard_get_delay_factor ()

```
gdouble
ags_soundcard_get_delay_factor (AgsSoundcard *soundcard);
```

Get current playback delay factor.

Parameters

soundcard	the AgsSoundcard

Returns

delay factor

Since: 3.0.0

ags_soundcard_get_absolute_delay ()

```
gdouble
ags_soundcard_get_absolute_delay (AgsSoundcard *soundcard);
```

Get current playback delay.

Parameters

soundcard	the AgsSoundcard

Returns

delay

Since: 3.0.0

ags_soundcard_get_delay ()

```
gdouble
ags_soundcard_get_delay (AgsSoundcard *soundcard);
```

Get current playback delay.

Parameters

soundcard	the AgsSoundcard

Returns

delay

Since: 3.0.0

ags_soundcard_get_attack ()

```
guint
ags_soundcard_get_attack (AgsSoundcard *soundcard);
```

Get current playback attack.

Parameters

soundcard	the AgsSoundcard

Returns

attack

Since: 3.0.0

ags_soundcard_get_delay_counter ()

```
guint
ags_soundcard_get_delay_counter (AgsSoundcard *soundcard);
```

Get current playback note offset.

Parameters

soundcard	the AgsSoundcard

Returns

offset

Since: 3.0.0

ags_soundcard_set_start_note_offset ()

```
void
ags_soundcard_set_start_note_offset (AgsSoundcard *soundcard,
                                     guint start_note_offset);
```

Set start playback note offset.

Parameters

soundcard	the AgsSoundcard
start_note_offset	the start note offset to set

Since: 3.0.0

ags_soundcard_get_start_note_offset ()

```
guint
ags_soundcard_get_start_note_offset (AgsSoundcard *soundcard);
```

Get start playback note offset.

Parameters

soundcard	the AgsSoundcard

Returns

the start note offset

Since: 3.0.0

ags_soundcard_set_note_offset ()

```
void
ags_soundcard_set_note_offset (AgsSoundcard *soundcard,
                               guint note_offset);
```

Set current playback note offset.

Parameters

soundcard	the AgsSoundcard
note_offset	the note offset to set

Since: 3.0.0

ags_soundcard_get_note_offset ()

```
guint
ags_soundcard_get_note_offset (AgsSoundcard *soundcard);
```

Get current playback note offset.

Parameters

soundcard	the AgsSoundcard

Returns

the current note offset

Since: 3.0.0

ags_soundcard_set_note_offset_absolute ()

```
void
ags_soundcard_set_note_offset_absolute
                                (AgsSoundcard *soundcard,
                                 guint note_offset);
```

Set current playback note offset.

Parameters

soundcard	the AgsSoundcard	
note_offset	the note offset to set	

Since: 3.0.0

ags_soundcard_get_note_offset_absolute ()

```
guint
ags_soundcard_get_note_offset_absolute
                                (AgsSoundcard *soundcard);
```

Get current playback note offset.

Parameters

soundcard	the AgsSoundcard	

Returns

offset

Since: 3.0.0

ags_soundcard_set_loop ()

```
void
ags_soundcard_set_loop (AgsSoundcard *soundcard,
                        guint loop_left,
                        guint loop_right,
                        gboolean do_loop);
```

Set loop parameters of *soundcard* .

Parameters

soundcard	the AgsSoundcard	
loop_left	loop position of region	
loop_right	loop position of region	
do_loop	if TRUE do loop, else don't loop	

Since: 3.0.0

ags_soundcard_get_loop ()

```
void
ags_soundcard_get_loop (AgsSoundcard *soundcard,
                        guint *loop_left,
                        guint *loop_right,
                        gboolean *do_loop);
```

Get loop parameters of *soundcard* .

Parameters

soundcard	the AgsSoundcard	
loop_left	return location of loop position's region.	*[out]*
loop_right	return location of loop position's region.	*[out]*
do_loop	return location of do loop.	*[out]*

Since: 3.0.0

ags_soundcard_get_loop_offset ()

```
guint
ags_soundcard_get_loop_offset (AgsSoundcard *soundcard);
```

Get current playback loop offset.

Parameters

soundcard	the AgsSoundcard	

Returns

offset

Since: 3.0.0

ags_soundcard_get_sub_block_count ()

```
guint
ags_soundcard_get_sub_block_count (AgsSoundcard *soundcard);
```

Get sub block count.

Parameters

soundcard	the AgsSoundcard	

Returns

the sub block count

Since: 3.0.0

ags_soundcard_trylock_sub_block ()

```
gboolean
ags_soundcard_trylock_sub_block (AgsSoundcard *soundcard,
                                 void *buffer,
                                 guint sub_block);
```

Trylock sub block.

Parameters

soundcard	the AgsSoundcard	
buffer	the buffer to lock	
sub_block	and its sub block	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_soundcard_unlock_sub_block ()

```
void
ags_soundcard_unlock_sub_block (AgsSoundcard *soundcard,
                                void *buffer,
                                guint sub_block);
```

Unlock sub block.

Parameters

soundcard	the AgsSoundcard	
buffer	the buffer to lock	
sub_block	and its sub block	

Since: 3.0.0

AGS_IS_SOUNDCARD()

```
#define AGS_IS_SOUNDCARD(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj), ↩
    AGS_TYPE_SOUNDCARD))
```

AGS_IS_SOUNDCARD_INTERFACE()

```
#define AGS_IS_SOUNDCARD_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SOUNDCARD))
```

AGS_SOUNDCARD()

```
#define AGS_SOUNDCARD(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOUNDCARD, AgsSoundcard))
```

AGS_SOUNDCARD_GET_INTERFACE()

```
#define AGS_SOUNDCARD_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SOUNDCARD, AgsSoundcardInterface))
```

AGS_SOUNDCARD_INTERFACE()

```
#define AGS_SOUNDCARD_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SOUNDCARD, AgsSoundcardInterface))
```

ags_soundcard_capability_get_type ()

```
GType
ags_soundcard_capability_get_type ();
```

ags_soundcard_format_get_type ()

```
GType
ags_soundcard_format_get_type ();
```

ags_soundcard_get_type ()

```
GType
ags_soundcard_get_type ();
```

Types and Values

AGS_SOUNDCARD_DEFAULT_DSP_CHANNELS

```
#define AGS_SOUNDCARD_DEFAULT_DSP_CHANNELS (2)
```

AGS_SOUNDCARD_MIN_DSP_CHANNELS

```
#define AGS_SOUNDCARD_MIN_DSP_CHANNELS (1)
```

AGS_SOUNDCARD_MAX_DSP_CHANNELS

```
#define AGS_SOUNDCARD_MAX_DSP_CHANNELS (64)
```

AGS_SOUNDCARD_DEFAULT_PCM_CHANNELS

```
#define AGS_SOUNDCARD_DEFAULT_PCM_CHANNELS (2)
```

AGS_SOUNDCARD_MIN_PCM_CHANNELS

```
#define AGS_SOUNDCARD_MIN_PCM_CHANNELS (1)
```

AGS_SOUNDCARD_MAX_PCM_CHANNELS

```
#define AGS_SOUNDCARD_MAX_PCM_CHANNELS (64)
```

AGS_SOUNDCARD_DEFAULT_SAMPLERATE

```
#define AGS_SOUNDCARD_DEFAULT_SAMPLERATE (48000.0)
```

AGS_SOUNDCARD_MIN_SAMPLERATE

```
#define AGS_SOUNDCARD_MIN_SAMPLERATE (8000.0)
```

AGS_SOUNDCARD_MAX_SAMPLERATE

```
#define AGS_SOUNDCARD_MAX_SAMPLERATE (5644800.0)
```

AGS_SOUNDCARD_DEFAULT_FORMAT

```
#define AGS_SOUNDCARD_DEFAULT_FORMAT (AGS_SOUNDCARD_SIGNED_16_BIT)
```

AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE

```
#define AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE (1024)
```

AGS_SOUNDCARD_MIN_BUFFER_SIZE

```
#define AGS_SOUNDCARD_MIN_BUFFER_SIZE (16)
```

AGS_SOUNDCARD_MAX_BUFFER_SIZE

```
#define AGS_SOUNDCARD_MAX_BUFFER_SIZE (16384)
```

AGS_SOUNDCARD_DEFAULT_BPM

```
#define AGS_SOUNDCARD_DEFAULT_BPM (120.0)
```

AGS_SOUNDCARD_DEFAULT_DELAY_FACTOR

```
#define AGS_SOUNDCARD_DEFAULT_DELAY_FACTOR (1.0 / 4.0)
```

AGS_SOUNDCARD_DEFAULT_JIFFIE

```
#define AGS_SOUNDCARD_DEFAULT_JIFFIE ((double) AGS_SOUNDCARD_DEFAULT_SAMPLERATE / (double)  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE)
```

AGS_SOUNDCARD_DEFAULT_TACT

```
#define AGS_SOUNDCARD_DEFAULT_TACT (1.0 / 1.0)
```

AGS_SOUNDCARD_DEFAULT_TACT_JIFFIE

```
#define AGS_SOUNDCARD_DEFAULT_TACT_JIFFIE (60.0 / AGS_SOUNDCARD_DEFAULT_BPM *  ↩
    AGS_SOUNDCARD_DEFAULT_TACT)
```

AGS_SOUNDCARD_DEFAULT_TACTRATE

```
#define AGS_SOUNDCARD_DEFAULT_TACTRATE (1.0 / AGS_SOUNDCARD_DEFAULT_TACT_JIFFIE)
```

AGS_SOUNDCARD_DEFAULT_SCALE

```
#define AGS_SOUNDCARD_DEFAULT_SCALE (1.0)
```

AGS_SOUNDCARD_DEFAULT_DELAY

```
#define AGS_SOUNDCARD_DEFAULT_DELAY ((60.0 * ((AGS_SOUNDCARD_DEFAULT_SAMPLERATE /  ↩
    AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE) / AGS_SOUNDCARD_DEFAULT_BPM)) * ((1.0 / 16.0) * (1.0  ↩
    / AGS_SOUNDCARD_DEFAULT_DELAY_FACTOR)))
```

AGS_SOUNDCARD_DEFAULT_PERIOD

```
#define AGS_SOUNDCARD_DEFAULT_PERIOD (2048.0)
```

AGS_SOUNDCARD_DEFAULT_LOOP_LEFT

```
#define AGS_SOUNDCARD_DEFAULT_LOOP_LEFT (0)
```

AGS_SOUNDCARD_DEFAULT_LOOP_RIGHT

```
#define AGS_SOUNDCARD_DEFAULT_LOOP_RIGHT (64)
```

AGS_SOUNDCARD_DEFAULT_OVERCLOCK

```
#define AGS_SOUNDCARD_DEFAULT_OVERCLOCK (2.0)
```

AGS_SOUNDCARD_DEFAULT_SUB_BLOCK_COUNT

```
#define AGS_SOUNDCARD_DEFAULT_SUB_BLOCK_COUNT (8)
```

enum AgsSoundcardFormat

AgsSoundcardFormat specifies the audio data representation to be used.

Members

AGS_SOUNDCARD_SIGNED_8_BIT	signed 8 bit raw pcm data
AGS_SOUNDCARD_SIGNED_16_BIT	signed 16 bit raw pcm data
AGS_SOUNDCARD_SIGNED_24_BIT	signed 24 bit raw pcm data
AGS_SOUNDCARD_SIGNED_32_BIT	signed 32 bit raw pcm data

AGS_SOUNDCARD_SIGNED_64_BIT	signed 64 bit raw pcm data
AGS_SOUNDCARD_FLOAT	float raw pcm data
AGS_SOUNDCARD_DOUBLE	double raw pcm data
AGS_SOUNDCARD_COMPLEX	complex au- dio data

enum AgsSoundcardCapability

Enum values to tell about capability.

Members

AGS_SOUNDCARD_CAPABILITY_PLAYBACK	capability play- back
AGS_SOUNDCARD_CAPABILITY_CAPTURE	capability cap- ture
AGS_SOUNDCARD_CAPABILITY_DUPLEX	capability du- plex

AGS_TYPE_SOUNDCARD

```
#define AGS_TYPE_SOUNDCARD                    (ags_soundcard_get_type())
```

AGS_TYPE_SOUNDCARD_CAPABILITY

```
#define AGS_TYPE_SOUNDCARD_CAPABILITY         (ags_soundcard_capability_get_type())
```

AGS_TYPE_SOUNDCARD_FORMAT

```
#define AGS_TYPE_SOUNDCARD_FORMAT             (ags_soundcard_format_get_type())
```

AgsSoundcard

```
typedef struct _AgsSoundcard AgsSoundcard;
```

struct AgsSoundcardInterface

```
struct AgsSoundcardInterface {
  GTypeInterface ginterface;

  void (*set_device)(AgsSoundcard *soundcard,
         gchar *card_id);
  gchar* (*get_device)(AgsSoundcard *soundcard);

  void (*pcm_info)(AgsSoundcard *soundcard, gchar *card_id,
       guint *channels_min, guint *channels_max,
       guint *rate_min, guint *rate_max,
       guint *buffer_size_min, guint *buffer_size_max,
       GError **error);

  guint (*get_capability)(AgsSoundcard *soundcard);

  void (*set_presets)(AgsSoundcard *soundcard,
          guint channels,
          guint rate,
          guint buffer_size,
          guint format);
  void (*get_presets)(AgsSoundcard *soundcard,
          guint *channels,
          guint *rate,
          guint *buffer_size,
          guint *format);

  void (*list_cards)(AgsSoundcard *soundcard,
         GList **card_id, GList **card_name);

  gboolean (*is_available)(AgsSoundcard *soundcard);

  gboolean (*is_starting)(AgsSoundcard *soundcard);
  gboolean (*is_playing)(AgsSoundcard *soundcard);
  gboolean (*is_recording)(AgsSoundcard *soundcard);

  gchar* (*get_uptime)(AgsSoundcard *soundcard);

  void (*play_init)(AgsSoundcard *soundcard,
        GError **error);
  void (*play)(AgsSoundcard *soundcard,
         GError **error);

  void (*record_init)(AgsSoundcard *soundcard,
          GError **error);
  void (*record)(AgsSoundcard *soundcard,
     GError **error);

  void (*stop)(AgsSoundcard *soundcard);

  void (*tic)(AgsSoundcard *soundcard);
  void (*offset_changed)(AgsSoundcard *soundcard,
       guint note_offset);

  void* (*get_buffer)(AgsSoundcard *soundcard);
  void* (*get_next_buffer)(AgsSoundcard *soundcard);

  void* (*get_prev_buffer)(AgsSoundcard *soundcard);

  void (*lock_buffer)(AgsSoundcard *soundcard,
          void *buffer);
```

```
  void (*unlock_buffer)(AgsSoundcard *soundcard,
      void *buffer);

  void (*set_bpm)(AgsSoundcard *soundcard,
      gdouble bpm);
  gdouble (*get_bpm)(AgsSoundcard *soundcard);

  void (*set_delay_factor)(AgsSoundcard *soundcard,
         gdouble delay_factor);
  gdouble (*get_delay_factor)(AgsSoundcard *soundcard);

  gdouble (*get_absolute_delay)(AgsSoundcard *soundcard);

  gdouble (*get_delay)(AgsSoundcard *soundcard);
  guint (*get_attack)(AgsSoundcard *soundcard);

  guint (*get_delay_counter)(AgsSoundcard *soundcard);

  void (*set_start_note_offset)(AgsSoundcard *soundcard,
        guint start_note_offset);
  guint (*get_start_note_offset)(AgsSoundcard *soundcard);

  void (*set_note_offset)(AgsSoundcard *soundcard,
        guint note_offset);
  guint (*get_note_offset)(AgsSoundcard *soundcard);

  void (*set_note_offset_absolute)(AgsSoundcard *soundcard,
           guint note_offset);
  guint (*get_note_offset_absolute)(AgsSoundcard *soundcard);

  void (*set_loop)(AgsSoundcard *soundcard,
       guint loop_left, guint loop_right,
       gboolean do_loop);
  void (*get_loop)(AgsSoundcard *soundcard,
       guint *loop_left, guint *loop_right,
       gboolean *do_loop);

  guint (*get_loop_offset)(AgsSoundcard *soundcard);

  guint (*get_sub_block_count)(AgsSoundcard *soundcard);

  gboolean (*trylock_sub_block)(AgsSoundcard *soundcard,
        void *buffer, guint sub_block);
  void (*unlock_sub_block)(AgsSoundcard *soundcard,
         void *buffer, guint sub_block);
};
```

Signal Details

The "offset-changed" signal

```
void
user_function (AgsSoundcard *soundcard,
               guint         note_offset,
               gpointer      user_data)
```

The ::offset-changed signal notifies about changed position within notation.

Parameters

soundcard	the AgsSoundcard object	
note_offset	new notation offset	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop" signal

```
void
user_function (AgsSoundcard *soundcard,
               gpointer      user_data)
```

The ::stop signal is emitted every stop of the soundcard. This notifies about a newly played buffer.

Parameters

soundcard	the AgsSoundcard object	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "tic" signal

```
void
user_function (AgsSoundcard *soundcard,
               gpointer      user_data)
```

The ::tic signal is emitted every tic of the soundcard. This notifies about a newly played buffer.

Parameters

soundcard	the AgsSoundcard object	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 38

AgsSoundServer

AgsSoundServer — access sound servers

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_sound_server_set_url ()
gchar *	ags_sound_server_get_url ()
void	ags_sound_server_set_ports ()
guint *	ags_sound_server_get_ports ()
void	ags_sound_server_set_soundcard ()
GList *	ags_sound_server_get_soundcard ()
void	ags_sound_server_set_sequencer ()
GList *	ags_sound_server_get_sequencer ()
GObject *	ags_sound_server_register_soundcard ()
void	ags_sound_server_unregister_soundcard ()
GObject *	ags_sound_server_register_sequencer ()
void	ags_sound_server_unregister_sequencer ()
#define	AGS_IS_SOUND_SERVER()
#define	AGS_IS_SOUND_SERVER_INTERFACE()
#define	AGS_SOUND_SERVER()
#define	AGS_SOUND_SERVER_GET_INTERFACE()
#define	AGS_SOUND_SERVER_INTERFACE()
GType	ags_sound_server_get_type ()

Types and Values

#define	AGS_TYPE_SOUND_SERVER
	AgsSoundServer
struct	AgsSoundServerInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsSoundServer
```

Includes

```
#include <ags/object/ags_sound_server.h>
```

Description

The AgsSoundServer interface gives you a unique access sound servers.

Functions

ags_sound_server_set_url ()

```
void
ags_sound_server_set_url (AgsSoundServer *sound_server,
                          gchar *url);
```

Sets the url of *sound_server* .

Parameters

sound_server	The AgsSoundServer	
url	the url to set	

Since: 3.0.0

ags_sound_server_get_url ()

```
gchar~*
ags_sound_server_get_url (AgsSoundServer *sound_server);
```

Gets the URL of *sound_server* .

Parameters

sound_server	the AgsSoundServer	

Returns

the URL as string

Since: 3.0.0

ags_sound_server_set_ports ()

```
void
ags_sound_server_set_ports (AgsSoundServer *sound_server,
                            guint *port,
                            guint port_count);
```

ags_sound_server_get_ports ()

```
guint~*
ags_sound_server_get_ports (AgsSoundServer *sound_server,
                            guint *port_count);
```

Gets the ports of *sound_server* .

Parameters

sound_server	the AgsSoundServer	
port_count	the number of ports returned	

Returns

the port as string NULL-terminated array

Since: 3.0.0

ags_sound_server_set_soundcard ()

```
void
ags_sound_server_set_soundcard (AgsSoundServer *sound_server,
                                gchar *client_uuid,
                                GList *soundcard);
```

Sets the soundcard at *client_uuid* .

Parameters

sound_server	The AgsSoundServer	
client_uuid	the location to fetch from	
soundcard	the soundcard to set.	*[element-type GObject][transfer full]*

Since: 3.0.0

ags_sound_server_get_soundcard ()

```
GList~*
ags_sound_server_get_soundcard (AgsSoundServer *sound_server,
                                gchar *client_uuid);
```

Gets the soundcard of *sound_server* associated with *client_uuid* .

Parameters

client_uuid	the client uuid	
sound_server	the AgsSoundServer	

Returns

the soundcard as GList.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_server_set_sequencer ()

```
void
ags_sound_server_set_sequencer (AgsSoundServer *sound_server,
                                gchar *client_uuid,
                                GList *sequencer);
```

Sets the sequencer at `client_uuid`.

Parameters

sound_server	The AgsSoundServer	
client_uuid	the location to fetch from	
sequencer	the sequencer to set.	*[element-type GObject][transfer full]*

Since: 3.0.0

ags_sound_server_get_sequencer ()

```
GList~*
ags_sound_server_get_sequencer (AgsSoundServer *sound_server,
                                gchar *client_uuid);
```

Gets the sequencer of `sound_server` associated with `client_uuid`.

Parameters

client_uuid	the client uuid	
sound_server	the AgsSoundServer	

Returns

the sequencer as GList.

[element-type GObject][transfer full]

Since: 3.0.0

ags_sound_server_register_soundcard ()

```
GObject~*
ags_sound_server_register_soundcard (AgsSoundServer *sound_server,
                                     gboolean is_output);
```

Fetches *soundcard* of *sound_server* .

Parameters

sound_server	The AgsSoundServer	
is_output	if TRUE the used as sink, else as source	

Returns

a new AgsSoundcard.

[transfer full]

Since: 3.0.0

ags_sound_server_unregister_soundcard ()

```
void
ags_sound_server_unregister_soundcard (AgsSoundServer *sound_server,
                                       GObject *soundcard);
```

Releases *soundcard* in *sound_server* .

Parameters

sound_server	The AgsSoundServer	
soundcard	the AgsSoundcard	

Since: 3.0.0

ags_sound_server_register_sequencer ()

```
GObject~*
ags_sound_server_register_sequencer (AgsSoundServer *sound_server,
                                     gboolean is_output);
```

Fetches *sequencer* of *sound_server* .

Parameters

sound_server	The AgsSoundServer	
is_output	if TRUE the used as sink, else as source	

Returns

a new AgsSequencer.

[transfer full]

Since: 3.0.0

ags_sound_server_unregister_sequencer ()

```
void
ags_sound_server_unregister_sequencer (AgsSoundServer *sound_server,
                                       GObject *sequencer);
```

Releases *sequencer* in *sound_server* .

Parameters

sound_server	The AgsSoundServer	
sequencer	the AgsSequencer	

Since: 3.0.0

AGS_IS_SOUND_SERVER()

```
#define AGS_IS_SOUND_SERVER(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SOUND_SERVER))
```

AGS_IS_SOUND_SERVER_INTERFACE()

```
#define AGS_IS_SOUND_SERVER_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SOUND_SERVER))
```

AGS_SOUND_SERVER()

```
#define AGS_SOUND_SERVER(obj)                       (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SOUND_SERVER, AgsSoundServer))
```

AGS_SOUND_SERVER_GET_INTERFACE()

```
#define AGS_SOUND_SERVER_GET_INTERFACE(obj)         (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SOUND_SERVER, AgsSoundServerInterface))
```

AGS_SOUND_SERVER_INTERFACE()

```
#define AGS_SOUND_SERVER_INTERFACE(vtable)          (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SOUND_SERVER, AgsSoundServerInterface))
```

ags_sound_server_get_type ()

```
GType
ags_sound_server_get_type ();
```

Types and Values

AGS_TYPE_SOUND_SERVER

```
#define AGS_TYPE_SOUND_SERVER                    (ags_sound_server_get_type())
```

AgsSoundServer

```
typedef struct _AgsSoundServer AgsSoundServer;
```

struct AgsSoundServerInterface

```
struct AgsSoundServerInterface {
  GTypeInterface ginterface;

  void (*set_url)(AgsSoundServer *sound_server,
      gchar *url);
  gchar* (*get_url)(AgsSoundServer *sound_server);

  void (*set_ports)(AgsSoundServer *sound_server,
        guint *ports, guint port_count);
  guint* (*get_ports)(AgsSoundServer *sound_server,
          guint *port_count);

  void (*set_soundcard)(AgsSoundServer *sound_server,
      gchar *client_uuid,
      GList *soundcard);
  GList* (*get_soundcard)(AgsSoundServer *sound_server,
        gchar *client_uuid);

  void (*set_sequencer)(AgsSoundServer *sound_server,
      gchar *client_uuid,
      GList *sequencer);
  GList* (*get_sequencer)(AgsSoundServer *sound_server,
        gchar *client_uuid);

  GObject* (*register_soundcard)(AgsSoundServer *sound_server,
         gboolean is_output);
  void (*unregister_soundcard)(AgsSoundServer *sound_server,
             GObject *soundcard);

  GObject* (*register_sequencer)(AgsSoundServer *sound_server,
         gboolean is_output);
  void (*unregister_sequencer)(AgsSoundServer *sound_server,
             GObject *sequencer);
};
```

Chapter 39

AgsTactable

AgsTactable — Unique tempo set or attributes access

Stability Level

Stable, unless otherwise indicated

Functions

guint64	ags_tactable_get_sequencer_duration ()
guint64	ags_tactable_get_notation_duration ()
guint64	ags_tactable_get_wave_duration ()
guint64	ags_tactable_get_midi_duration ()
gdouble	ags_tactable_get_bpm ()
gdouble	ags_tactable_get_tact ()
void	ags_tactable_change_sequencer_duration ()
void	ags_tactable_change_notation_duration ()
void	ags_tactable_change_wave_duration ()
void	ags_tactable_change_midi_duration ()
void	ags_tactable_change_bpm ()
void	ags_tactable_change_tact ()
#define	AGS_IS_TACTABLE()
#define	AGS_IS_TACTABLE_INTERFACE()
#define	AGS_TACTABLE()
#define	AGS_TACTABLE_GET_INTERFACE()
#define	AGS_TACTABLE_INTERFACE()
GType	ags_tactable_get_type ()

Signals

void	change-bpm	Run Last
void	change-midi-duration	Run Last
void	change-notation-duration	Run Last
void	change-sequencer-duration	Run Last
void	change-tact	Run Last

void	change-wave-duration	Run Last

Types and Values

#define	AGS_TYPE_TACTABLE
	AgsTactable
struct	AgsTactableInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsTactable
```

Includes

```
#include <ags/object/ags_tactable.h>
```

Description

The AgsTactable interface gives you a unique access to modify tempo.

Functions

ags_tactable_get_sequencer_duration ()

```
guint64
ags_tactable_get_sequencer_duration (AgsTactable *tactable);
```

Get sequencer duration.

Parameters

tactable	the AgsTactable

Returns

the sequencer duration

Since: 3.0.0

ags_tactable_get_notation_duration ()

```
guint64
ags_tactable_get_notation_duration (AgsTactable *tactable);
```

Get notation duration.

Parameters

tactable	the AgsTactable

Returns

the notation duration

Since: 3.0.0

ags_tactable_get_wave_duration ()

```
guint64
ags_tactable_get_wave_duration (AgsTactable *tactable);
```

Get wave duration.

Parameters

tactable	the AgsTactable

Returns

the wave duration

Since: 3.0.0

ags_tactable_get_midi_duration ()

```
guint64
ags_tactable_get_midi_duration (AgsTactable *tactable);
```

Get midi duration.

Parameters

tactable	the AgsTactable

Returns

the midi duration

Since: 3.0.0

ags_tactable_get_bpm ()

```
gdouble
ags_tactable_get_bpm (AgsTactable *tactable);
```

Get bpm.

Parameters

tactable	the AgsTactable

Returns

the bpm

Since: 3.0.0

ags_tactable_get_tact ()

```
gdouble
ags_tactable_get_tact (AgsTactable *tactable);
```

Get tact.

Parameters

tactable	the AgsTactable

Returns

the tact

Since: 3.0.0

ags_tactable_change_sequencer_duration ()

```
void
ags_tactable_change_sequencer_duration
                                (AgsTactable *tactable,
                                 guint64 sequencer_duration);
```

Change sequencer duration.

Parameters

tactable	the AgsTactable
sequencer_duration	the duration

Since: 3.0.0

ags_tactable_change_notation_duration ()

```
void
ags_tactable_change_notation_duration (AgsTactable *tactable,
                                       guint64 notation_duration);
```

Change notation duration.

Parameters

tactable	the AgsTactable	
notation_duration	the duration	

Since: 3.0.0

ags_tactable_change_wave_duration ()

```
void
ags_tactable_change_wave_duration (AgsTactable *tactable,
                                   guint64 wave_duration);
```

Change wave duration.

Parameters

tactable	the AgsTactable	
wave_duration	the duration	

Since: 3.0.0

ags_tactable_change_midi_duration ()

```
void
ags_tactable_change_midi_duration (AgsTactable *tactable,
                                   guint64 midi_duration);
```

Change midi duration.

Parameters

tactable	the AgsTactable	
midi_duration	the duration	

Since: 3.0.0

ags_tactable_change_bpm ()

```
void
ags_tactable_change_bpm (AgsTactable *tactable,
                         gdouble new_bpm,
                         gdouble old_bpm);
```

Change bpm.

Parameters

tactable	the AgsTactable	
new_bpm	the new bpm	
old_bpm	the old bpm	

Since: 3.0.0

ags_tactable_change_tact ()

```
void
ags_tactable_change_tact (AgsTactable *tactable,
                          gdouble new_tact,
                          gdouble old_tact);
```

Change tact.

Parameters

tactable	the AgsTactable	
new_tact	the new tact	
old_tact	the old tact	

Since: 3.0.0

AGS_IS_TACTABLE()

```
#define AGS_IS_TACTABLE(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_TACTABLE))
```

AGS_IS_TACTABLE_INTERFACE()

```
#define AGS_IS_TACTABLE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_TACTABLE))
```

AGS_TACTABLE()

```
#define AGS_TACTABLE(obj)                       (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TACTABLE, AgsTactable))
```

AGS_TACTABLE_GET_INTERFACE()

```
#define AGS_TACTABLE_GET_INTERFACE(obj)         (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_TACTABLE, AgsTactableInterface))
```

AGS_TACTABLE_INTERFACE()

```
#define AGS_TACTABLE_INTERFACE(vtable)          (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_TACTABLE, AgsTactableInterface))
```

ags_tactable_get_type ()

```
GType
ags_tactable_get_type ();
```

Types and Values

AGS_TYPE_TACTABLE

```
#define AGS_TYPE_TACTABLE                    (ags_tactable_get_type())
```

AgsTactable

```
typedef struct _AgsTactable AgsTactable;
```

struct AgsTactableInterface

```
struct AgsTactableInterface {
  GTypeInterface ginterface;

  guint64 (*get_sequencer_duration)(AgsTactable *tactable);
  guint64 (*get_notation_duration)(AgsTactable *tactable);
  guint64 (*get_wave_duration)(AgsTactable *tactable);
  guint64 (*get_midi_duration)(AgsTactable *tactable);

  gdouble (*get_bpm)(AgsTactable *tactable);
  gdouble (*get_tact)(AgsTactable *tactable);

  void (*change_sequencer_duration)(AgsTactable *tactable, guint64 sequencer_duration);
  void (*change_notation_duration)(AgsTactable *tactable, guint64 notation_duration);
  void (*change_wave_duration)(AgsTactable *tactable, guint64 wave_duration);
  void (*change_midi_duration)(AgsTactable *tactable, guint64 midi_duration);

  void (*change_bpm)(AgsTactable *tactable, gdouble new_bpm, gdouble old_bpm);
  void (*change_tact)(AgsTactable *tactable, gdouble new_tact, gdouble old_tact);
};
```

Signal Details

The "change-bpm" signal

```
void
user_function (AgsTactable *tactable,
               double       new_bpm,
               double       old_bpm,
               gpointer     user_data)
```

The ::change-bpm signal notifies about changed bpm.

Parameters

tactable	the AgsTactable object	
new_bpm	new BPM	
old_bpm	old BPM	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "change-midi-duration" signal

```
void
user_function (AgsTactable *tactable,
               guint64      midi_duration,
               gpointer     user_data)
```

The ::change-midi-duration signal notifies about changed duration of midi.

Parameters

tactable	the AgsTactable object	
midi_duration	new duration	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "change-notation-duration" signal

```
void
user_function (AgsTactable *tactable,
               guint64      notation_duration,
               gpointer     user_data)
```

The ::change-notation-duration signal notifies about changed duration of notation.

Parameters

tactable	the AgsTactable object	
notation_duration	new duration	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "change-sequencer-duration" signal

```
void
user_function (AgsTactable *tactable,
               guint64      sequencer_duration,
               gpointer     user_data)
```

The ::change-sequencer-duration signal notifies about changed duration of sequencer.

Parameters

tactable	the AgsTactable object	
sequencer_duration	new duration	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "change-tact" signal

```
void
user_function (AgsTactable *tactable,
               double       new_tact,
               double       old_tact,
               gpointer     user_data)
```

The ::change-tact signal notifies about changed tact.

Parameters

tactable	the AgsTactable object	
new_tact	new tact	
old_tact	old tact	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "change-wave-duration" signal

```
void
user_function (AgsTactable *tactable,
               guint64      wave_duration,
               gpointer     user_data)
```

The ::change-wave-duration signal notifies about changed duration of wave.

Parameters

tactable	the AgsTactable object	
wave_duration	new duration	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Part IV

XML Input/Output - Persist project

Chapter 40

AgsFile

AgsFile — read/write XML file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FILE_GET_OBJ_MUTEX()
void	ags_file_set_filename ()
gchar *	ags_file_get_filename ()
void	ags_file_set_encoding ()
gchar *	ags_file_get_encoding ()
void	ags_file_set_audio_format ()
gchar *	ags_file_get_audio_format ()
void	ags_file_set_audio_encoding ()
gchar *	ags_file_get_audio_encoding ()
void	ags_file_set_xml_doc ()
xmlDoc *	ags_file_get_xml_doc ()
gchar *	ags_file_str2md5 ()
void	ags_file_add_id_ref ()
GObject *	ags_file_find_id_ref_by_node ()
GObject *	ags_file_find_id_ref_by_xpath ()
GObject *	ags_file_find_id_ref_by_reference ()
void	ags_file_add_lookup ()
void	ags_file_add_launch ()
xmlDoc *	ags_file_xml_new_doc ()
xmlNode *	ags_file_xml_new_node ()
xmlNode *	ags_file_xml_get_root_element ()
void	ags_file_xml_set_root_element ()
void	ags_file_xml_add_child ()
gchar *	ags_file_xml_get_node_name ()
xmlNode *	ags_file_xml_get_node_parent ()
xmlNode *	ags_file_xml_get_node_next ()
xmlNode *	ags_file_xml_get_node_children ()
gchar *	ags_file_xml_get_prop ()

void	ags_file_xml_set_prop ()
gchar *	ags_file_xml_get_content ()
void	ags_file_xml_set_content ()
void	ags_file_open ()
void	ags_file_open_from_data ()
void	ags_file_rw_open ()
void	ags_file_open_filename ()
void	ags_file_close ()
void	ags_file_write ()
void	ags_file_write_concurrent ()
void	ags_file_write_resolve ()
void	ags_file_read ()
void	ags_file_read_resolve ()
void	ags_file_read_start ()
void	ags_file_read_config ()
void	ags_file_write_config ()
void	ags_file_read_application_context ()
void	ags_file_write_application_context ()
AgsFile *	ags_file_new ()
#define	AGS_FILE()
#define	AGS_FILE_CLASS()
#define	AGS_FILE_GET_CLASS()
#define	AGS_IS_FILE()
#define	AGS_IS_FILE_CLASS()
GType	ags_file_flags_get_type ()
GType	ags_file_get_type ()

Properties

char *	audio-encoding	Read / Write
char *	audio-format	Read / Write
char *	encoding	Read / Write
char *	filename	Read / Write
gpointer	xml-doc	Read / Write

Signals

void	open	Run Last
void	open-from-data	Run Last
void	read	Run Last
void	read-resolve	Run Last
void	read-start	Run Last
void	rw-open	Run Last
void	write	Run Last
void	write-concurrent	Run Last
void	write-resolve	Run Last

Types and Values

#define	AGS_FILE_DEFAULT_APP_ENCODING

#define	AGS_FILE_DEFAULT_ENCODING
#define	AGS_FILE_DEFAULT_DTD
#define	AGS_FILE_CHECKSUM_LENGTH
#define	AGS_FILE_DEFAULT_AUDIO_FORMAT
#define	AGS_FILE_DEFAULT_AUDIO_ENCODING
#define	AGS_FILE_CHARSET_CONVERTER_MAX_STRING_LENGTH
#define	AGS_FILE_CHARSET_CONVERTER_MAX_CONTENT_LENGTH
enum	AgsFileFlags
#define	AGS_FILE_ERROR
enum	AgsFileError
#define	AGS_TYPE_FILE
#define	AGS_TYPE_FILE_FLAGS
struct	AgsFile
struct	AgsFileClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsFileFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsFile
```

Includes

```
#include <ags/file/ags_file.h>
```

Description

The AgsFile is an object to read or write files using XML. It is the persisting layer of Advanced Gtk+ Sequencer.

The application encoding used is either C.UTF-8 or taken from LANG environment variable.

The following functions convert between encoding and app_encoding:

- ags_file_xml_new_doc()
- ags_file_xml_new_node()
- ags_file_xml_get_prop()
- ags_file_xml_set_prop()
- ags_file_xml_get_content()
- ags_file_xml_set_content()

Functions

AGS_FILE_GET_OBJ_MUTEX()

```
#define AGS_FILE_GET_OBJ_MUTEX(obj) (&(((AgsFile *) obj)->obj_mutex))
```

ags_file_set_filename ()

```
void
ags_file_set_filename (AgsFile *file,
                       gchar *filename);
```

Set *filename* of *file* .

Parameters

file	the AgsFile	
filename	the filename	

Since: 3.6.17

ags_file_get_filename ()

```
gchar~*
ags_file_get_filename (AgsFile *file);
```

Get filename of *file* .

Parameters

file	the AgsFile	

Returns

the filename.

[transfer full]

Since: 3.6.17

ags_file_set_encoding ()

```
void
ags_file_set_encoding (AgsFile *file,
                       gchar *encoding);
```

Set *encoding* of *file* .

Parameters

file	the AgsFile	
encoding	the encoding	

Since: 3.6.17

ags_file_get_encoding ()

```
gchar~*
ags_file_get_encoding (AgsFile *file);
```

Get encoding of *file* .

Parameters

file	the AgsFile

Returns

the encoding.

[transfer full]

Since: 3.6.17

ags_file_set_audio_format ()

```
void
ags_file_set_audio_format (AgsFile *file,
                           gchar *audio_format);
```

Set *audio_format* of *file* .

Parameters

file	the AgsFile
audio_format	the audio format

Since: 3.6.17

ags_file_get_audio_format ()

```
gchar~*
ags_file_get_audio_format (AgsFile *file);
```

Get audio format of *file* .

Parameters

file	the AgsFile

Returns

the audio format.

[transfer full]

Since: 3.6.17

ags_file_set_audio_encoding ()

```
void
ags_file_set_audio_encoding (AgsFile *file,
                             gchar *audio_encoding);
```

Set *audio_encoding* of *file* .

Parameters

file	the AgsFile	
audio_encoding	the audio encoding	

Since: 3.6.17

ags_file_get_audio_encoding ()

```
gchar~*
ags_file_get_audio_encoding (AgsFile *file);
```

Get audio encoding of *file* .

Parameters

file	the AgsFile	

Returns

the audio encoding.

[transfer full]

Since: 3.6.17

ags_file_set_xml_doc ()

```
void
ags_file_set_xml_doc (AgsFile *file,
                      xmlDoc *xml_doc);
```

Set *xml_doc* of *file* .

Parameters

file	the AgsFile	
xml_doc	the xmlDoc.	*[transfer full]*

Since: 3.6.17

ags_file_get_xml_doc ()

```
xmlDoc~*
ags_file_get_xml_doc (AgsFile *file);
```

Get xml doc of *file* .

Parameters

file	the AgsFile

Returns

the xmlDoc.

[transfer none]

Since: 3.6.17

ags_file_str2md5 ()

```
gchar~*
ags_file_str2md5 (gchar *content,
                  guint content_length);
```

Compute MD5 sums of a buffer.

Parameters

content	the string buffer
content_length	the length of the string

Returns

the md5 checksum.

[transfer full]

Since: 3.0.0

ags_file_add_id_ref ()

```
void
ags_file_add_id_ref (AgsFile *file,
                     GObject *id_ref);
```

Adds *id_ref* to *file* .

Parameters

file	the *AgsFile*	
id_ref	a reference	

Since: 3.0.0

ags_file_find_id_ref_by_node ()

```
GObject~*
ags_file_find_id_ref_by_node (AgsFile *file,
                              xmlNode *node);
```

Find a reference by its XML node.

Parameters

file	the *AgsFile*	
node	a XML node	

Returns

the matching GObject.

[transfer full]

Since: 3.0.0

ags_file_find_id_ref_by_xpath ()

```
GObject~*
ags_file_find_id_ref_by_xpath (AgsFile *file,
                               gchar *xpath);
```

Lookup a reference by *xpath* .

Parameters

file	the AgsFile	
xpath	a XPath expression	

Returns

the matching GObject.

[transfer full]

Since: 3.0.0

ags_file_find_id_ref_by_reference ()

```
GObject~*
ags_file_find_id_ref_by_reference (AgsFile *file,
                                   gpointer ref);
```

Find a reference matching *ref* .

Parameters

file	the AgsFile	
ref	a gpointer	

Returns

the matching GObject.

[transfer full]

Since: 3.0.0

ags_file_add_lookup ()

```
void
ags_file_add_lookup (AgsFile *file,
                     GObject *file_lookup);
```

Add *file_lookup* for later invoking.

Parameters

file	the AgsFile	
file_lookup	a AgsFileLookup	

Since: 3.0.0

ags_file_add_launch ()

```
void
ags_file_add_launch (AgsFile *file,
                     GObject *file_launch);
```

Add *file_launch* for later invoking.

Parameters

file	the AgsFile	
file_launch	a AgsFileLaunch	

Since: 3.0.0

ags_file_xml_new_doc ()

```
xmlDoc~*
ags_file_xml_new_doc (AgsFile *file,
                      gchar *version);
```

Create xmlDoc.

Parameters

file	the AgsFile	
version	the version	

Returns

the newly created xmlDoc.

[transfer full]

Since: 4.0.0

ags_file_xml_new_node ()

```
xmlNode~*
ags_file_xml_new_node (AgsFile *file,
                       xmlNs *xml_namespace,
                       gchar *node_name);
```

Create xmlNode.

Parameters

file	the AgsFile	
xml_namespace	the XML namespace	
node_name	the XML node name	

Returns

the newly created xmlNode.

[transfer full]

Since: 4.0.0

ags_file_xml_get_root_element ()

```
xmlNode~*
ags_file_xml_get_root_element (AgsFile *file,
                               xmlDoc *doc);
```

Get root element of `doc` .

Parameters

file	the AgsFile	
doc	the xmlDoc	

Returns

the root element of xmlDoc.

[transfer none]

Since: 4.0.0

ags_file_xml_set_root_element ()

```
void
ags_file_xml_set_root_element (AgsFile *file,
                               xmlDoc *doc,
                               xmlNode *root_node);
```

Set root element `root_node` of `doc` .

Parameters

file	the AgsFile	
doc	the xmlDoc	
root_node	the xmlNode as root node.	*[transfer full]*

Since: 4.0.0

ags_file_xml_add_child ()

```
void
ags_file_xml_add_child (AgsFile *file,
                        xmlNode *parent,
                        xmlNode *child);
```

Set root element `root_node` of `doc` .

Parameters

file	the AgsFile	
parent	the parent xmlNode	
child	the child xmlNode.	*[transfer full]*

Since: 4.0.0

ags_file_xml_get_node_name ()

```
gchar~*
ags_file_xml_get_node_name (AgsFile *file,
                            xmlNode *node);
```

Get node name of `node` .

Parameters

file	the AgsFile	
node	the xmlNode	

Returns

the node name of xmlNode.

[transfer full]

Since: 4.0.0

ags_file_xml_get_node_parent ()

```
xmlNode~*
ags_file_xml_get_node_parent (AgsFile *file,
                              xmlNode *node);
```

ags_file_xml_get_node_next ()

```
xmlNode~*
ags_file_xml_get_node_next (AgsFile *file,
                            xmlNode *node);
```

ags_file_xml_get_node_children ()

```
xmlNode~*
ags_file_xml_get_node_children (AgsFile *file,
                                xmlNode *node);
```

ags_file_xml_get_prop ()

```
gchar~*
ags_file_xml_get_prop (AgsFile *file,
                       xmlNode *node,
                       gchar *prop_name);
```

Get property of `node` .

Parameters

file	the AgsFile	
node	the xmlNode	
prop_name	property name	

Returns

the property of xmlNode.

[transfer full]

Since: 4.0.0

ags_file_xml_set_prop ()

```
void
ags_file_xml_set_prop (AgsFile *file,
                       xmlNode *node,
                       gchar *prop_name,
                       gchar *prop_value);
```

Set property of *node* .

Parameters

file	the AgsFile	
node	the xmlNode	
prop_name	property name	
prop_value	property value	

Since: 4.0.0

ags_file_xml_get_content ()

```
gchar~*
ags_file_xml_get_content (AgsFile *file,
                          xmlNode *node);
```

Get content of *node* .

Parameters

file	the AgsFile	
node	the xmlNode	

Returns

the content of xmlNode.

[transfer full]

Since: 4.0.0

ags_file_xml_set_content ()

```
void
ags_file_xml_set_content (AgsFile *file,
                          xmlNode *node,
                          gchar *content,
                          gboolean is_cdata);
```

Set content of *node* .

Parameters

file	the AgsFile	
node	the xmlNode	
content	the content	
is_cdata	if content is CDATA	

Since: 4.0.0

ags_file_open ()

```
void
ags_file_open (AgsFile *file,
               GError **error);
```

Opens the file specified by :filename property.

Parameters

file	the AgsFile	
error	the return location for errors	

Since: 3.0.0

ags_file_open_from_data ()

```
void
ags_file_open_from_data (AgsFile *file,
                         gchar *data,
                         guint length,
                         GError **error);
```

Opens the file provided by *data* .

Parameters

file	the AgsFile	
data	a buffer containing the XML document	
length	the buffer length	
error	the return location for errors	

Since: 3.0.0

ags_file_rw_open ()

```
void
ags_file_rw_open (AgsFile *file,
                  gboolean create,
                  GError **error);
```

Opens the file specified by :filename property in read-write mode.

Parameters

file	the AgsFile	
create	if TRUE create the file as needed	
error	the return location for errors	

Since: 3.0.0

ags_file_open_filename ()

```
void
ags_file_open_filename (AgsFile *file,
                        gchar *filename);
```

Opens the file specified by *filename* property.

Parameters

file	the AgsFile	
filename	a path	

Since: 3.0.0

ags_file_close ()

```
void
ags_file_close (AgsFile *file);
```

Closes *file*.

Parameters

file	the AgsFile	

Since: 3.0.0

ags_file_write ()

```
void
ags_file_write (AgsFile *file);
```

Write the XML document to disk.

Parameters

file	the AgsFile	

Since: 3.0.0

ags_file_write_concurrent ()

```
void
ags_file_write_concurrent (AgsFile *file);
```

ags_file_write_resolve ()

```
void
ags_file_write_resolve (AgsFile *file);
```

Resolve references to XPath expressions.

Parameters

file	the AgsFile

Since: 3.0.0

ags_file_read ()

```
void
ags_file_read (AgsFile *file);
```

Read XML document from disk.

Parameters

file	the AgsFile

Since: 3.0.0

ags_file_read_resolve ()

```
void
ags_file_read_resolve (AgsFile *file);
```

ags_file_read_start ()

```
void
ags_file_read_start (AgsFile *file);
```

Update or start the application.

Parameters

file	the AgsFile

Since: 3.0.0

ags_file_read_config ()

```
void
ags_file_read_config (AgsFile *file,
                      xmlNode *node,
                      GObject **ags_config);
```

ags_file_write_config ()

```
void
ags_file_write_config (AgsFile *file,
                       xmlNode *parent,
                       GObject *ags_config);
```

ags_file_read_application_context ()

```
void
ags_file_read_application_context (AgsFile *file,
                                   xmlNode *node,
                                   GObject **application_context);
```

ags_file_write_application_context ()

```
void
ags_file_write_application_context (AgsFile *file,
                                    xmlNode *parent,
                                    GObject *application_context);
```

ags_file_new ()

```
AgsFile~*
ags_file_new ();
```

Creates an AgsFile

Returns

a new AgsFile

Since: 3.0.0

AGS_FILE()

```
#define AGS_FILE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_FILE, ↩
    AgsFile))
```

AGS_FILE_CLASS()

```
#define AGS_FILE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_FILE, ↩
    AgsFileClass))
```

AGS_FILE_GET_CLASS()

```
#define AGS_FILE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj), AGS_TYPE_FILE,  ↩
    AgsFileClass))
```

AGS_IS_FILE()

```
#define AGS_IS_FILE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_FILE))
```

AGS_IS_FILE_CLASS()

```
#define AGS_IS_FILE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_FILE))
```

ags_file_flags_get_type ()

```
GType
ags_file_flags_get_type ();
```

ags_file_get_type ()

```
GType
ags_file_get_type (void);
```

Types and Values

AGS_FILE_DEFAULT_APP_ENCODING

```
#define AGS_FILE_DEFAULT_APP_ENCODING "C.UTF-8"
```

AGS_FILE_DEFAULT_ENCODING

```
#define AGS_FILE_DEFAULT_ENCODING "UTF-8"
```

AGS_FILE_DEFAULT_DTD

```
#define AGS_FILE_DEFAULT_DTD "ags_file.dtd"
```

AGS_FILE_CHECKSUM_LENGTH

```
#define AGS_FILE_CHECKSUM_LENGTH (32)
```

AGS_FILE_DEFAULT_AUDIO_FORMAT

```
#define AGS_FILE_DEFAULT_AUDIO_FORMAT "raw"
```

AGS_FILE_DEFAULT_AUDIO_ENCODING

```
#define AGS_FILE_DEFAULT_AUDIO_ENCODING "base64"
```

AGS_FILE_CHARSET_CONVERTER_MAX_STRING_LENGTH

```
#define AGS_FILE_CHARSET_CONVERTER_MAX_STRING_LENGTH (8192)
```

AGS_FILE_CHARSET_CONVERTER_MAX_CONTENT_LENGTH

```
#define AGS_FILE_CHARSET_CONVERTER_MAX_CONTENT_LENGTH (8388608) // max 8 MB
```

enum AgsFileFlags

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

Members

AGS_FILE_READ	opened file for reading
AGS_FILE_READ_AUDIO_SIGNAL	read audio signal
AGS_FILE_READ_EMBEDDED_AUDIO	read embedded audio data
AGS_FILE_WRITE	opened file for writing
AGS_FILE_WRITE_AUDIO_SIGNAL	write audio signal

AGS_FILE_WRITE_EMBEDDED_AUDIO	write embedded audio data

AGS_FILE_ERROR

```
#define AGS_FILE_ERROR (ags_file_error_quark())
```

enum AgsFileError

Enum values to indicated failures to used with GError.

Members

AGS_FILE_ERROR_PARSER_FAILURE	the parser failed

AGS_TYPE_FILE

```
#define AGS_TYPE_FILE                (ags_file_get_type())
```

AGS_TYPE_FILE_FLAGS

```
#define AGS_TYPE_FILE_FLAGS          (ags_file_flags_get_type())
```

struct AgsFile

```
struct AgsFile;
```

struct AgsFileClass

```
struct AgsFileClass {
  GObjectClass gobject;

  void (*open)(AgsFile *file,
         GError **error);
  void (*open_from_data)(AgsFile *file,
       gchar *data, guint length,
       GError **error);
  void (*rw_open)(AgsFile *file,
      gboolean create,
      GError **error);

  void (*write)(AgsFile *file);
```

```
  void (*write_concurrent)(AgsFile *file);
  void (*write_resolve)(AgsFile *file);

  void (*read)(AgsFile *file);
  void (*read_resolve)(AgsFile *file);
  void (*read_start)(AgsFile *file);
};
```

Property Details

The "audio-encoding" property

```
  "audio-encoding"            char~*
```

The encoding to use for embedding audio data.

Owner: AgsFile

Flags: Read / Write

Default value: "base64"

Since: 3.0.0

The "audio-format" property

```
  "audio-format"              char~*
```

The format of embedded audio data.

Owner: AgsFile

Flags: Read / Write

Default value: "raw"

Since: 3.0.0

The "encoding" property

```
  "encoding"                  char~*
```

The charset encoding to use.

Owner: AgsFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "filename" property

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

The assigned filename to open and read from.

Owner: AgsFile

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "xml-doc" property

```
"xml-doc"                  gpointer
```

The assigned xml-doc.

Owner: AgsFile

Flags: Read / Write

Since: 3.0.0

Signal Details

The "open" signal

```
void
user_function (AgsFile *file,
               gpointer error,
               gpointer user_data)
```

Open *file* with appropriate filename.

Parameters

file	the AgsFile	
error	a GError pointer to return error	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "open-from-data" signal

```
void
user_function (AgsFile *file,
               char    *data,
               guint    length,
               gpointer error,
               gpointer user_data)
```

Open *file* from a buffer containing the file.

Parameters

file	the AgsFile	
data	the buffer containing the file	
length	the buffer length	

error	a GError pointer to return error	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "read" signal

```
void
user_function (AgsFile *file,
               gpointer user_data)
```

Read a XML document from disk with specified filename.

Parameters

file	the AgsFile	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "read-resolve" signal

```
void
user_function (AgsFile *file,
               gpointer user_data)
```

Resolve XPath expressions to their counterpart the newly created instances refering to.

Parameters

file	the AgsFile	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "read-start" signal

```
void
user_function (AgsFile *file,
               gpointer user_data)
```

Hook after reading XML document to update or start the application.

Parameters

file	the AgsFile	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "rw-open" signal

```
void
user_function (AgsFile *agsfile,
               gboolean arg1,
               gpointer arg2,
               gpointer user_data)
```

Flags: Run Last

The "write" signal

```
void
user_function (AgsFile *file,
               gpointer user_data)
```

Write XML Document to disk.

Parameters

file	the AgsFile	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "write-concurrent" signal

```
void
user_function (AgsFile *agsfile,
               gpointer user_data)
```

Flags: Run Last

The "write-resolve" signal

```
void
user_function (AgsFile *file,
               gpointer user_data)
```

Resolve references and generate thus XPath expressions just before writing to disk.

Parameters

file	the AgsFile	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 41

AgsFileIdRef

AgsFileIdRef — file id reference

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FILE_ID_REF_GET_OBJ_MUTEX()
#define	AGS_FILE_RESOLVE()
void	(*AgsFileResolve) ()
void	ags_file_id_ref_resolved ()
AgsFileIdRef *	ags_file_id_ref_new ()
#define	AGS_FILE_ID_REF()
#define	AGS_FILE_ID_REF_CLASS()
#define	AGS_FILE_ID_REF_GET_CLASS()
#define	AGS_IS_FILE_ID_REF()
#define	AGS_IS_FILE_ID_REF_CLASS()
GType	ags_file_id_ref_get_type ()

Properties

GObject *	file	Read / Write
gpointer	node	Read / Write
gpointer	reference	Read / Write
char *	xpath	Read / Write

Signals

void	resolved	Run Last

Types and Values

#define	AGS_FILE_ID_REF_SERIALIZE_DATA
#define	AGS_FILE_ID_REF_RESOLVE_DATA
#define	AGS_TYPE_FILE_ID_REF
struct	AgsFileIdRef
struct	AgsFileIdRefClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFileIdRef
```

Includes

```
#include <ags/file/ags_file_id_ref.h>
```

Description

The AgsFileIdRef referes a XML node by its ID. It's used to perform XPath lookup.

Functions

AGS_FILE_ID_REF_GET_OBJ_MUTEX()

```
#define AGS_FILE_ID_REF_GET_OBJ_MUTEX(obj) (&(((AgsFileIdRef *) obj)->obj_mutex))
```

AGS_FILE_RESOLVE()

```
#define AGS_FILE_RESOLVE(f)           ((AgsFileResolve)(f))
```

AgsFileResolve ()

```
void
(*AgsFileResolve) (void);
```

ags_file_id_ref_resolved ()

```
void
ags_file_id_ref_resolved (AgsFileIdRef *file_id_ref);
```

Notify about resolved reference.

Parameters

file_id_ref	the AgsFileIdRef

Since: 3.0.0

ags_file_id_ref_new ()

```
AgsFileIdRef~*
ags_file_id_ref_new ();
```

Creates an AgsFileIdRef

Returns

a new AgsFileIdRef

Since: 3.0.0

AGS_FILE_ID_REF()

```
#define AGS_FILE_ID_REF(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FILE_ID_REF, AgsFileIdRef))
```

AGS_FILE_ID_REF_CLASS()

```
#define AGS_FILE_ID_REF_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_FILE_ID_REF, AgsFileIdRef))
```

AGS_FILE_ID_REF_GET_CLASS()

```
#define AGS_FILE_ID_REF_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_FILE_ID_REF, AgsFileIdRefClass))
```

AGS_IS_FILE_ID_REF()

```
#define AGS_IS_FILE_ID_REF(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FILE_ID_REF))
```

AGS_IS_FILE_ID_REF_CLASS()

```
#define AGS_IS_FILE_ID_REF_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FILE_ID_REF))
```

ags_file_id_ref_get_type ()

```
GType
ags_file_id_ref_get_type ();
```

Types and Values

AGS_FILE_ID_REF_SERIALIZE_DATA

```
#define AGS_FILE_ID_REF_SERIALIZE_DATA "ags-file-id-ref-serizalize-data"
```

AGS_FILE_ID_REF_RESOLVE_DATA

```
#define AGS_FILE_ID_REF_RESOLVE_DATA "ags-file-id-ref-resolve-data"
```

AGS_TYPE_FILE_ID_REF

```
#define AGS_TYPE_FILE_ID_REF                (ags_file_id_ref_get_type())
```

struct AgsFileIdRef

```
struct AgsFileIdRef;
```

struct AgsFileIdRefClass

```
struct AgsFileIdRefClass {
  GObjectClass gobject;

  void (*resolved)(AgsFileIdRef *file_id_ref);
};
```

Property Details

The "file" property

```
  "file"                      GObject~*
```

The AgsFile this AgsFileIdRef belongs to.

Owner: AgsFileIdRef

Flags: Read / Write

Since: 3.0.0

The "node" property

```
  "node"                      gpointer
```

The assigned xmlNode being referred by this AgsFileIdRef.

Owner: AgsFileIdRef

Flags: Read / Write

Since: 3.0.0

The "reference" property

```
"reference"                gpointer
```

The object referred by this AgsFileIdRef.

Owner: AgsFileIdRef

Flags: Read / Write

Since: 3.0.0

The "xpath" property

```
"xpath"                    char~*
```

The XPath short-cut that can be used with this AgsFileIdRef.

Owner: AgsFileIdRef

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Signal Details

The "resolved" signal

```
void
user_function (AgsFileIdRef *file_id_ref,
               gpointer      user_data)
```

Signal ::resolved to notify about resolved :reference.

Parameters

file_id_ref	the AgsFileIdRef	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 42

AgsFileLink

AgsFileLink — link objects read of file.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FILE_LINK_GET_OBJ_MUTEX()
void	ags_file_link_set_filename ()
gchar *	ags_file_link_get_filename ()
void	ags_file_link_set_data ()
gchar *	ags_file_link_get_data ()
AgsFileLink *	ags_file_link_new ()
#define	AGS_FILE_LINK()
#define	AGS_FILE_LINK_CLASS()
#define	AGS_FILE_LINK_GET_CLASS()
#define	AGS_IS_FILE_LINK()
#define	AGS_IS_FILE_LINK_CLASS()
GType	ags_file_link_get_type ()

Properties

char *	data	Read / Write
char *	filename	Read / Write

Types and Values

#define	AGS_FILE_LINK_DEFAULT_VERSION
#define	AGS_FILE_LINK_DEFAULT_BUILD_ID
#define	AGS_TYPE_FILE_LINK
struct	AgsFileLink

struct	AgsFileLinkClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFileLink
```

Implemented Interfaces

AgsFileLink implements AgsPlugin.

Includes

```
#include <ags/file/ags_file_link.h>
```

Description

The AgsFileLink links read objects of file.

Functions

AGS_FILE_LINK_GET_OBJ_MUTEX()

```
#define AGS_FILE_LINK_GET_OBJ_MUTEX(obj) (&(((AgsFileLink *) obj)->obj_mutex))
```

ags_file_link_set_filename ()

```
void
ags_file_link_set_filename (AgsFileLink *file_link,
                            gchar *filename);
```

Set *filename* of *file_link* .

Parameters

file_link	the AgsFileLink	
filename	the filename	

Since: 3.6.17

ags_file_link_get_filename ()

```
gchar~*
ags_file_link_get_filename (AgsFileLink *file_link);
```

Get filename of *file_link* .

Parameters

file_link	the AgsFileLink

Returns

the filename.

[transfer full]

Since: 3.6.17

ags_file_link_set_data ()

```
void
ags_file_link_set_data (AgsFileLink *file_link,
                        gchar *data);
```

Set *data* of *file_link* .

Parameters

file_link	the AgsFileLink
data	the data

Since: 3.6.17

ags_file_link_get_data ()

```
gchar~*
ags_file_link_get_data (AgsFileLink *file_link);
```

Get data of *file_link* .

Parameters

file_link	the AgsFileLink

Returns

the data.

[transfer full]

Since: 3.6.17

ags_file_link_new ()

```
AgsFileLink~*
ags_file_link_new ();
```

Creates an AgsFileLink

Returns

a new AgsFileLink

Since: 3.0.0

AGS_FILE_LINK()

```
#define AGS_FILE_LINK(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FILE_LINK, AgsFileLink))
```

AGS_FILE_LINK_CLASS()

```
#define AGS_FILE_LINK_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_FILE_LINK, AgsFileLink))
```

AGS_FILE_LINK_GET_CLASS()

```
#define AGS_FILE_LINK_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_FILE_LINK, AgsFileLinkClass))
```

AGS_IS_FILE_LINK()

```
#define AGS_IS_FILE_LINK(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FILE_LINK))
```

AGS_IS_FILE_LINK_CLASS()

```
#define AGS_IS_FILE_LINK_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FILE_LINK))
```

ags_file_link_get_type ()

```
GType
ags_file_link_get_type ();
```

Types and Values

AGS_FILE_LINK_DEFAULT_VERSION

```
#define AGS_FILE_LINK_DEFAULT_VERSION "0.7.0"
```

AGS_FILE_LINK_DEFAULT_BUILD_ID

```
#define AGS_FILE_LINK_DEFAULT_BUILD_ID "CEST 13-10-2015 15:53"
```

AGS_TYPE_FILE_LINK

```
#define AGS_TYPE_FILE_LINK                (ags_file_link_get_type())
```

struct AgsFileLink

```
struct AgsFileLink;
```

struct AgsFileLinkClass

```
struct AgsFileLinkClass {
  GObjectClass gobject;
};
```

Property Details

The "data" property

```
  "data"                    char~*
```

The data this AgsFileLink contains.

Owner: AgsFileLink

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "filename" property

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

The filename this AgsFileLink refers.

Owner: AgsFileLink

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Chapter 43

AgsFileLookup

AgsFileLookup — resolve objects read of file.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FILE_LOOKUP_GET_OBJ_MUTEX()
GList *	ags_file_lookup_find_by_node ()
GList *	ags_file_lookup_find_by_reference ()
void	ags_file_lookup_resolve ()
AgsFileLookup *	ags_file_lookup_new ()
#define	AGS_FILE_LOOKUP()
#define	AGS_FILE_LOOKUP_CLASS()
#define	AGS_FILE_LOOKUP_GET_CLASS()
#define	AGS_IS_FILE_LOOKUP()
#define	AGS_IS_FILE_LOOKUP_CLASS()
GType	ags_file_lookup_get_type ()

Properties

GObject *	file	Read / Write
gpointer	node	Read / Write
gpointer	reference	Read / Write

Signals

void	resolve	Run Last

Types and Values

#define	AGS_TYPE_FILE_LOOKUP
struct	AgsFileLookup
struct	AgsFileLookupClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFileLookup
```

Includes

```
#include <ags/file/ags_file_lookup.h>
```

Description

The AgsFileLookup resolve objects of file.

Functions

AGS_FILE_LOOKUP_GET_OBJ_MUTEX()

```
#define AGS_FILE_LOOKUP_GET_OBJ_MUTEX(obj) (&(((AgsFileLookup *) obj)->obj_mutex))
```

ags_file_lookup_find_by_node ()

```
GList~*
ags_file_lookup_find_by_node (GList *file_lookup,
                              xmlNode *node);
```

Find AgsFileLookup by xmlNode

Parameters

file_lookup	the GList containing AgsFileLookup.	*[element-type Ags.FileLookup][transfer none]*
node	a xmlNode	

Returns

The matching list containing AgsFileLookup if found otherwise NULL.

[element-type Ags.FileLookup][transfer none]

Since: 3.0.0

ags_file_lookup_find_by_reference ()

```
GList~*
ags_file_lookup_find_by_reference (GList *file_lookup,
                                   gpointer ref);
```

Find AgsFileLookup by *ref*

Parameters

file_lookup	the GList containing AgsFileLookup.	*[element-type Ags.FileLookup][transfer none]*
ref	a gpointer	

Returns

The matching list containing AgsFileLookup if found otherwise NULL.

[element-type Ags.FileLookup][transfer none]

Since: 3.0.0

ags_file_lookup_resolve ()

```
void
ags_file_lookup_resolve (AgsFileLookup *file_lookup);
```

The ::resolve signal.

Parameters

file_lookup	the AgsFileLookup	

Since: 3.0.0

ags_file_lookup_new ()

```
AgsFileLookup~*
ags_file_lookup_new ();
```

Creates an AgsFileLookup

Returns

a new AgsFileLookup

Since: 3.0.0

AGS_FILE_LOOKUP()

```
#define AGS_FILE_LOOKUP(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FILE_LOOKUP, AgsFileLookup))
```

AGS_FILE_LOOKUP_CLASS()

```
#define AGS_FILE_LOOKUP_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FILE_LOOKUP, AgsFileLookupClass))
```

AGS_FILE_LOOKUP_GET_CLASS()

```
#define AGS_FILE_LOOKUP_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FILE_LOOKUP, AgsFileLookupClass))
```

AGS_IS_FILE_LOOKUP()

```
#define AGS_IS_FILE_LOOKUP(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FILE_LOOKUP))
```

AGS_IS_FILE_LOOKUP_CLASS()

```
#define AGS_IS_FILE_LOOKUP_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FILE_LOOKUP))
```

ags_file_lookup_get_type ()

```
GType
ags_file_lookup_get_type (void);
```

Types and Values

AGS_TYPE_FILE_LOOKUP

```
#define AGS_TYPE_FILE_LOOKUP                (ags_file_lookup_get_type())
```

struct AgsFileLookup

```
struct AgsFileLookup;
```

struct AgsFileLookupClass

```
struct AgsFileLookupClass {
  GObjectClass gobject;

  void (*resolve)(AgsFileLookup *file_lookup);
};
```

Property Details

The "file" property

```
  "file"                      GObject~*
```

The file it is assigned with.

Owner: AgsFileLookup

Flags: Read / Write

The "node" property

```
  "node"                      gpointer
```

The node it is assigned with.

Owner: AgsFileLookup

Flags: Read / Write

The "reference" property

```
  "reference"                 gpointer
```

The reference it is assigned with.

Owner: AgsFileLookup

Flags: Read / Write

Signal Details

The "resolve" signal

```
void
user_function (AgsFileLookup *agsfilelookup,
               gpointer       user_data)
```

Flags: Run Last

Chapter 44

AgsFileLaunch

AgsFileLaunch — launch objects read of file.

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_FILE_LAUNCH_GET_OBJ_MUTEX()
void	ags_file_launch_start ()
AgsFileLaunch *	ags_file_launch_new ()
#define	AGS_FILE_LAUNCH()
#define	AGS_FILE_LAUNCH_CLASS()
#define	AGS_FILE_LAUNCH_GET_CLASS()
#define	AGS_IS_FILE_LAUNCH()
#define	AGS_IS_FILE_LAUNCH_CLASS()
GType	ags_file_launch_get_type ()

Properties

GObject *	file	Read / Write
gpointer	node	Read / Write
gpointer	reference	Read / Write

Signals

void	start	Run Last

Types and Values

#define	AGS_TYPE_FILE_LAUNCH
struct	AgsFileLaunch
struct	AgsFileLaunchClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsFileLaunch
```

Includes

```
#include <ags/file/ags_file_launch.h>
```

Description

The AgsFileLaunch launches read objects of file.

Functions

AGS_FILE_LAUNCH_GET_OBJ_MUTEX()

```
#define AGS_FILE_LAUNCH_GET_OBJ_MUTEX(obj) (&(((AgsFileLaunch *) obj)->obj_mutex))
```

ags_file_launch_start ()

```
void
ags_file_launch_start (AgsFileLaunch *file_launch);
```

Start AgsFileLaunch to fulfill a task.

Parameters

file_launch	the AgsFileLaunch

Since: 3.0.0

ags_file_launch_new ()

```
AgsFileLaunch~*
ags_file_launch_new ();
```

Create a new instance of AgsFileLaunch

Returns

the new AgsFileLaunch

Since: 3.0.0

AGS_FILE_LAUNCH()

```
#define AGS_FILE_LAUNCH(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FILE_LAUNCH, AgsFileLaunch))
```

AGS_FILE_LAUNCH_CLASS()

```
#define AGS_FILE_LAUNCH_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_FILE_LAUNCH, AgsFileLaunchClass))
```

AGS_FILE_LAUNCH_GET_CLASS()

```
#define AGS_FILE_LAUNCH_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_FILE_LAUNCH, AgsFileLaunchClass))
```

AGS_IS_FILE_LAUNCH()

```
#define AGS_IS_FILE_LAUNCH(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FILE_LAUNCH))
```

AGS_IS_FILE_LAUNCH_CLASS()

```
#define AGS_IS_FILE_LAUNCH_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FILE_LAUNCH))
```

ags_file_launch_get_type ()

```
GType
ags_file_launch_get_type (void);
```

Types and Values

AGS_TYPE_FILE_LAUNCH

```
#define AGS_TYPE_FILE_LAUNCH                (ags_file_launch_get_type())
```

struct AgsFileLaunch

```
struct AgsFileLaunch;
```

struct AgsFileLaunchClass

```
struct AgsFileLaunchClass {
  GObjectClass gobject;

  void (*start)(AgsFileLaunch *file_launch);
};
```

Property Details

The "file" property

```
  "file"                      GObject~*
```

The AgsFile this AgsFileLaunch belongs to.

Owner: AgsFileLaunch

Flags: Read / Write

Since: 3.0.0

The "node" property

```
  "node"                      gpointer
```

The assigned xmlNode being referred by this AgsFileLaunch.

Owner: AgsFileLaunch

Flags: Read / Write

Since: 3.0.0

The "reference" property

```
  "reference"                 gpointer
```

The object referred by this AgsFileLaunch.

Owner: AgsFileLaunch

Flags: Read / Write

Since: 3.0.0

Signal Details

The "start" signal

```
void
user_function (AgsFileLaunch *file_id_ref,
               gpointer       user_data)
```

Signal ::start to notify about start :reference.

Parameters

file_id_ref	the AgsFileLaunch	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Part V

Thread - Multi-threaded tree

Chapter 45

AgsConcurrencyProvider

AgsConcurrencyProvider — Concurrency provider interface

Stability Level

Stable, unless otherwise indicated

Functions

AgsThread *	ags_concurrency_provider_get_main_loop ()
void	ags_concurrency_provider_set_main_loop ()
AgsTaskLauncher *	ags_concurrency_provider_get_task_launcher ()
void	ags_concurrency_provider_set_task_launcher ()
AgsThreadPool *	ags_concurrency_provider_get_thread_pool ()
void	ags_concurrency_provider_set_thread_pool ()
GList *	ags_concurrency_provider_get_worker ()
void	ags_concurrency_provider_set_worker ()
#define	AGS_CONCURRENCY_PROVIDER()
#define	AGS_CONCURRENCY_PROVIDER_GET_INTERFACE()
#define	AGS_CONCURRENCY_PROVIDER_INTERFACE()
#define	AGS_IS_CONCURRENCY_PROVIDER()
#define	AGS_IS_CONCURRENCY_PROVIDER_INTERFACE()
GType	ags_concurrency_provider_get_type ()

Types and Values

#define	AGS_TYPE_CONCURRENCY_PROVIDER
	AgsConcurrencyProvider
struct	AgsConcurrencyProviderInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsConcurrencyProvider
```

Known Implementations

AgsConcurrencyProvider is implemented by AgsServerApplicationContext and AgsThreadApplicationContext.

Includes

```
#include <ags/thread/ags_concurrency_provider.h>
```

Description

The AgsConcurrencyProvider gives you unique access to threads.

Functions

ags_concurrency_provider_get_main_loop ()

```
AgsThread~*
ags_concurrency_provider_get_main_loop
                                (AgsConcurrencyProvider *concurrency_provider);
```

Get main loop of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider

Returns

the AgsThread implementing AgsMainLoop.

[transfer full]

Since: 3.0.0

ags_concurrency_provider_set_main_loop ()

```
void
ags_concurrency_provider_set_main_loop
                                (AgsConcurrencyProvider *concurrency_provider,
                                 AgsThread *main_loop);
```

Set main loop of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	
main_loop	the AgsThread implementing AgsMainLoop	

Since: 3.0.0

ags_concurrency_provider_get_task_launcher ()

```
AgsTaskLauncher~*
ags_concurrency_provider_get_task_launcher
                                (AgsConcurrencyProvider *concurrency_provider);
```

Get task launcher of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	

Returns

the AgsTaskLauncher.

[transfer full]

Since: 3.0.0

ags_concurrency_provider_set_task_launcher ()

```
void
ags_concurrency_provider_set_task_launcher
                                (AgsConcurrencyProvider *concurrency_provider,
                                 AgsTaskLauncher *task_launcher);
```

Set task launcher of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	
task_launcher	the AgsTaskLauncher	

Since: 3.0.0

ags_concurrency_provider_get_thread_pool ()

```
AgsThreadPool~*
ags_concurrency_provider_get_thread_pool
                                (AgsConcurrencyProvider *concurrency_provider);
```

Get thread pool of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	

Returns

the AgsThreadPool.

[transfer full]

Since: 3.0.0

ags_concurrency_provider_set_thread_pool ()

```
void
ags_concurrency_provider_set_thread_pool
                                (AgsConcurrencyProvider *concurrency_provider,
                                 AgsThreadPool *thread_pool);
```

Set thread pool of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	
thread_pool	the AgsThreadPool	

Since: 3.0.0

ags_concurrency_provider_get_worker ()

```
GList~*
ags_concurrency_provider_get_worker (AgsConcurrencyProvider *concurrency_provider);
```

Get workers of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	

Returns

the GList containing workers.

[element-type Ags.WorkerThread][transfer full]

Since: 3.0.0

ags_concurrency_provider_set_worker ()

```
void
ags_concurrency_provider_set_worker (AgsConcurrencyProvider *concurrency_provider,
                                     GList *worker);
```

Set workers of application context.

Parameters

concurrency_provider	the AgsConcurrencyProvider	
worker	the GList containing workers.	*[element-type Ags.WorkerThread]*

Since: 3.0.0

AGS_CONCURRENCY_PROVIDER()

```
#define AGS_CONCURRENCY_PROVIDER(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj), ↩
    AGS_TYPE_CONCURRENCY_PROVIDER, AgsConcurrencyProvider))
```

AGS_CONCURRENCY_PROVIDER_GET_INTERFACE()

```
#define AGS_CONCURRENCY_PROVIDER_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE(( ↩
    obj), AGS_TYPE_CONCURRENCY_PROVIDER, AgsConcurrencyProviderInterface))
```

AGS_CONCURRENCY_PROVIDER_INTERFACE()

```
#define AGS_CONCURRENCY_PROVIDER_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable), ↩
    AGS_TYPE_CONCURRENCY_PROVIDER, AgsConcurrencyProviderInterface))
```

AGS_IS_CONCURRENCY_PROVIDER()

```
#define AGS_IS_CONCURRENCY_PROVIDER(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj), ↩
    AGS_TYPE_CONCURRENCY_PROVIDER))
```

AGS_IS_CONCURRENCY_PROVIDER_INTERFACE()

```
#define AGS_IS_CONCURRENCY_PROVIDER_INTERFACE(vtable)    (G_TYPE_CHECK_CLASS_TYPE((vtable), ↩
    AGS_TYPE_CONCURRENCY_PROVIDER))
```

ags_concurrency_provider_get_type ()

```
GType
ags_concurrency_provider_get_type ();
```

Types and Values

AGS_TYPE_CONCURRENCY_PROVIDER

```
#define AGS_TYPE_CONCURRENCY_PROVIDER                    (ags_concurrency_provider_get_type ↩
    ())
```

AgsConcurrencyProvider

```
typedef struct _AgsConcurrencyProvider AgsConcurrencyProvider;
```

struct AgsConcurrencyProviderInterface

```
struct AgsConcurrencyProviderInterface {
  GTypeInterface ginterface;

  AgsThread* (*get_main_loop)(AgsConcurrencyProvider *concurrency_provider);
  void (*set_main_loop)(AgsConcurrencyProvider *concurrency_provider,
      AgsThread *main_loop);

  AgsTaskLauncher* (*get_task_launcher)(AgsConcurrencyProvider *concurrency_provider);
  void (*set_task_launcher)(AgsConcurrencyProvider *concurrency_provider,
          AgsTaskLauncher *task_launcher);

  AgsThreadPool* (*get_thread_pool)(AgsConcurrencyProvider *concurrency_provider);
  void (*set_thread_pool)(AgsConcurrencyProvider *concurrency_provider,
        AgsThreadPool *thread_pool);

  GList* (*get_worker)(AgsConcurrencyProvider *concurrency_provider);
  void (*set_worker)(AgsConcurrencyProvider *concurrency_provider,
         GList *worker);
};
```

Chapter 46

AgsDestroyWorker

AgsDestroyWorker — destroy worker

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_DESTROY_ENTRY()
void	(*AgsDestroyFunc) ()
AgsDestroyEntry *	ags_destroy_entry_alloc ()
void	ags_destroy_worker_add ()
AgsDestroyWorker *	ags_destroy_worker_get_instance ()
AgsDestroyWorker *	ags_destroy_worker_new ()
#define	AGS_DESTROY_WORKER()
#define	AGS_DESTROY_WORKER_CLASS()
#define	AGS_DESTROY_WORKER_GET_CLASS()
#define	AGS_IS_DESTROY_WORKER()
#define	AGS_IS_DESTROY_WORKER_CLASS()
GType	ags_destroy_worker_get_type ()

Types and Values

struct	AgsDestroyEntry
#define	AGS_TYPE_DESTROY_WORKER
struct	AgsDestroyWorker
struct	AgsDestroyWorkerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsThread
    &#x2570;&#x2500;&#x2500; AgsWorkerThread
        &#x2570;&#x2500;&#x2500; AgsDestroyWorker
```

Implemented Interfaces

AgsDestroyWorker implements AgsConnectable.

Includes

```
#include <ags/worker/ags_destroy_worker.h>
```

Description

The AgsDestroyWorker does non-realtime work. You might want give it the responsibility to destroy your objects.

Functions

AGS_DESTROY_ENTRY()

```
#define AGS_DESTROY_ENTRY(ptr) ((AgsDestroyEntry *) (ptr))
```

AgsDestroyFunc ()

```
void
(*AgsDestroyFunc) (gpointer ptr);
```

ags_destroy_entry_alloc ()

```
AgsDestroyEntry~*
ags_destroy_entry_alloc (gpointer ptr,
                         AgsDestroyFunc destroy_func);
```

Allocated a destroy entry.

Parameters

ptr	a pointer	
destroy_func	the `ptr` 's destroy function.	*[scope call]*

Returns

the allocated AgsDestroyEntry.

[type gpointer][transfer none]

Since: 3.0.0

ags_destroy_worker_add ()

```
void
ags_destroy_worker_add (AgsDestroyWorker *destroy_worker,
                        gpointer ptr,
                        AgsDestroyFunc destroy_func);
```

Add *ptr* for destruction using *destroy_func* .

Parameters

destroy_worker	the AgsDestroyWorker.	*[type gpointer]*
ptr	the gpointer to destroy	
destroy_func	the AgsDestroyFunc.	*[scope call]*

Since: 3.0.0

ags_destroy_worker_get_instance ()

```
AgsDestroyWorker~*
ags_destroy_worker_get_instance ();
```

Get your destroy worker instance.

Returns

the AgsDestroyWorker instance.

[transfer none]

Since: 3.0.0

ags_destroy_worker_new ()

```
AgsDestroyWorker~*
ags_destroy_worker_new ();
```

Create a new AgsDestroyWorker.

Returns

the new AgsDestroyWorker

Since: 3.0.0

AGS_DESTROY_WORKER()

```
#define AGS_DESTROY_WORKER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_DESTROY_WORKER, AgsDestroyWorker))
```

AGS_DESTROY_WORKER_CLASS()

```
#define AGS_DESTROY_WORKER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_DESTROY_WORKER, AgsDestroyWorkerClass))
```

AGS_DESTROY_WORKER_GET_CLASS()

```
#define AGS_DESTROY_WORKER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_DESTROY_WORKER, AgsDestroyWorkerClass))
```

AGS_IS_DESTROY_WORKER()

```
#define AGS_IS_DESTROY_WORKER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_DESTROY_WORKER))
```

AGS_IS_DESTROY_WORKER_CLASS()

```
#define AGS_IS_DESTROY_WORKER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_DESTROY_WORKER))
```

ags_destroy_worker_get_type ()

```
GType
ags_destroy_worker_get_type ();
```

Types and Values

struct AgsDestroyEntry

```
struct AgsDestroyEntry {
  gpointer ptr;

  void (*destroy_func)(gpointer ptr);
};
```

The struct to specify an entry to destroy.

Members

gpointer *ptr*;	the entry to destroy

destroy_func ()	the destroy function

AGS_TYPE_DESTROY_WORKER

```
#define AGS_TYPE_DESTROY_WORKER                (ags_destroy_worker_get_type())
```

struct AgsDestroyWorker

```
struct AgsDestroyWorker;
```

struct AgsDestroyWorkerClass

```
struct AgsDestroyWorkerClass {
  AgsWorkerThreadClass worker_thread;
};
```

Chapter 47

AgsGenericMainLoop

AgsGenericMainLoop — generic loop

Stability Level

Stable, unless otherwise indicated

Functions

AgsGenericMainLoop *	ags_generic_main_loop_new ()
#define	AGS_GENERIC_MAIN_LOOP()
#define	AGS_GENERIC_MAIN_LOOP_CLASS()
#define	AGS_GENERIC_MAIN_LOOP_GET_CLASS()
#define	AGS_IS_GENERIC_MAIN_LOOP()
#define	AGS_IS_GENERIC_MAIN_LOOP_CLASS()
GType	ags_generic_main_loop_get_type ()

Types and Values

#define	AGS_GENERIC_MAIN_LOOP_DEFAULT_JIFFIE
#define	AGS_TYPE_GENERIC_MAIN_LOOP
struct	AgsGenericMainLoop
struct	AgsGenericMainLoopClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsThread
    &#x2570;&#x2500;&#x2500; AgsGenericMainLoop
```

Implemented Interfaces

AgsGenericMainLoop implements AgsConnectable and AgsMainLoop.

Includes

```
#include <ags/thread/ags_generic_main_loop.h>
```

Description

The AgsGenericMainLoop is suitable as AgsMainLoop and does generic processing.

Functions

ags_generic_main_loop_new ()

```
AgsGenericMainLoop~*
ags_generic_main_loop_new ();
```

Create a new AgsGenericMainLoop.

Returns

the new AgsGenericMainLoop

Since: 3.0.0

AGS_GENERIC_MAIN_LOOP()

```
#define AGS_GENERIC_MAIN_LOOP(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_GENERIC_MAIN_LOOP, AgsGenericMainLoop))
```

AGS_GENERIC_MAIN_LOOP_CLASS()

```
#define AGS_GENERIC_MAIN_LOOP_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_GENERIC_MAIN_LOOP, AgsGenericMainLoopClass))
```

AGS_GENERIC_MAIN_LOOP_GET_CLASS()

```
#define AGS_GENERIC_MAIN_LOOP_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_GENERIC_MAIN_LOOP, AgsGenericMainLoopClass))
```

AGS_IS_GENERIC_MAIN_LOOP()

```
#define AGS_IS_GENERIC_MAIN_LOOP(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_GENERIC_MAIN_LOOP))
```

AGS_IS_GENERIC_MAIN_LOOP_CLASS()

```
#define AGS_IS_GENERIC_MAIN_LOOP_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_GENERIC_MAIN_LOOP))
```

ags_generic_main_loop_get_type ()

```
GType
ags_generic_main_loop_get_type ();
```

Types and Values

AGS_GENERIC_MAIN_LOOP_DEFAULT_JIFFIE

```
#define AGS_GENERIC_MAIN_LOOP_DEFAULT_JIFFIE (AGS_THREAD_DEFAULT_JIFFIE)
```

AGS_TYPE_GENERIC_MAIN_LOOP

```
#define AGS_TYPE_GENERIC_MAIN_LOOP                (ags_generic_main_loop_get_type())
```

struct AgsGenericMainLoop

```
struct AgsGenericMainLoop;
```

struct AgsGenericMainLoopClass

```
struct AgsGenericMainLoopClass {
  AgsThreadClass thread;
};
```

Chapter 48

AgsMessageDelivery

AgsMessageDelivery — Message delivery singleton

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MESSAGE_DELIVERY_GET_OBJ_MUTEX()
void	ags_message_delivery_set_message_queue ()
GList *	ags_message_delivery_get_message_queue ()
void	ags_message_delivery_add_message_queue ()
void	ags_message_delivery_remove_message_queue ()
GList *	ags_message_delivery_find_sender_namespace ()
GList *	ags_message_delivery_find_recipient_namespace ()
void	ags_message_delivery_add_message_envelope ()
void	ags_message_delivery_remove_message_envelope ()
GList *	ags_message_delivery_find_sender ()
GList *	ags_message_delivery_find_recipient ()
GList *	ags_message_delivery_query_message ()
AgsMessageDelivery *	ags_message_delivery_get_instance ()
AgsMessageDelivery *	ags_message_delivery_new ()
#define	AGS_IS_MESSAGE_DELIVERY()
#define	AGS_IS_MESSAGE_DELIVERY_CLASS()
#define	AGS_MESSAGE_DELIVERY()
#define	AGS_MESSAGE_DELIVERY_CLASS()
#define	AGS_MESSAGE_DELIVERY_GET_CLASS()
GType	ags_message_delivery_get_type ()

Types and Values

#define	AGS_TYPE_MESSAGE_DELIVERY
struct	AgsMessageDelivery
struct	AgsMessageDeliveryClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsMessageDelivery
```

Includes

```
#include <ags/thread/ags_message_delivery.h>
```

Description

The AgsMessageDelivery acts as messages passing system. Obtain the singleton instance by calling ags_message_delivery_get_instance()

To query the message queue for message envelopes there are functions like:

- ags_message_delivery_find_sender()
- ags_message_delivery_find_recipient()
- ags_message_delivery_query_message()

Make sure to cleanup the queue after you queried it.

ags_message_delivery_remove_message_envelope()

You instatiate a message queue with `sender-namespace` and `recipient-namespace`.

- ags_message_queue_new()
- ags_message_queue_set_recipient_namespace()

Then you subscribe to a message queue by calling

- ags_message_delivery_add_message_queue()

If you want to send a message, you first check if the AgsMessageQueue is available. Then you can add a message envelope by calling:

- ags_message_delivery_add_message_envelope()

Functions

AGS_MESSAGE_DELIVERY_GET_OBJ_MUTEX()

```
#define AGS_MESSAGE_DELIVERY_GET_OBJ_MUTEX(obj) (&(((AgsMessageDelivery *) obj)->obj_mutex) ↩
    )
```

ags_message_delivery_set_message_queue ()

```
void
ags_message_delivery_set_message_queue
                                (AgsMessageDelivery *message_delivery,
                                 GList *message_queue);
```

Set *message_queue* of *message_delivery* .

Parameters

message_delivery	the AgsMessageDelivery	
message_queue	the GList containing AgsMessageQueue.	*[element-type Ags.MessageQueue][transfer full]*

Since: 3.6.16

ags_message_delivery_get_message_queue ()

```
GList~*
ags_message_delivery_get_message_queue
                                (AgsMessageDelivery *message_delivery);
```

Get message queue of *message_delivery* .

Parameters

message_delivery	the AgsMessageDelivery	

Returns

the GList containing AgsMessageQueue.

[element-type Ags.MessageQueue][transfer none]

Since: 3.6.16

ags_message_delivery_add_message_queue ()

```
void
ags_message_delivery_add_message_queue
                                (AgsMessageDelivery *message_delivery,
                                 GObject *message_queue);
```

Add *message_queue* to *message_delivery* .

Parameters

message_delivery	the AgsMessageDelivery	
message_queue	the AgsMessageQueue	

Since: 3.0.0

ags_message_delivery_remove_message_queue ()

```
void
ags_message_delivery_remove_message_queue
                                (AgsMessageDelivery *message_delivery,
                                 GObject *message_queue);
```

Remove *message_queue* to *message_delivery* .

Parameters

message_delivery	the AgsMessageDelivery	
message_queue	the AgsMessageQueue	

Since: 3.0.0

ags_message_delivery_find_sender_namespace ()

```
GList~*
ags_message_delivery_find_sender_namespace
                                (AgsMessageDelivery *message_delivery,
                                 gchar *sender_namespace);
```

Find sender namespace of AgsMessageQueue in *message_delivery* .

Parameters

message_delivery	the AgsMessageDelivery to search	
sender_namespace	the sender namespace as string to find	

Returns

the matching GList containing AgsMessageQueue or NULL.

[element-type Ags.MessageQueue][transfer full]

Since: 3.0.0

ags_message_delivery_find_recipient_namespace ()

```
GList~*
ags_message_delivery_find_recipient_namespace
                                (AgsMessageDelivery *message_delivery,
                                 gchar *recipient_namespace);
```

Find recipient namespace of AgsMessageQueue in *message_delivery* .

Parameters

message_delivery	the AgsMessageDelivery to search	
recipient_namespace	the recipient namespace as string to find	

Returns

the matching GList containing AgsMessageQueue or NULL.

[element-type Ags.MessageQueue][transfer full]

Since: 3.0.0

ags_message_delivery_add_message_envelope ()

```
void
ags_message_delivery_add_message_envelope
                                (AgsMessageDelivery *message_delivery,
                                 gchar *sender_namespace,
                                 GObject *message_envelope);
```

Add *message* to an AgsMessageQueue specified by *sender_namespace* .

Parameters

message_delivery	the AgsMessageDelivery	
sender_namespace	the sender namespace as string	
message_envelope	the AgsMessageEnvelope	

Since: 3.0.0

ags_message_delivery_remove_message_envelope ()

```
void
ags_message_delivery_remove_message_envelope
                                (AgsMessageDelivery *message_delivery,
                                 gchar *sender_namespace,
                                 GObject *message_envelope);
```

Remove *message* from an AgsMessageQueue specified by *sender_namespace* .

Parameters

message_delivery	the AgsMessageDelivery	
sender_namespace	the sender namespace as string	
message_envelope	the AgsMessageEnvelope	

Since: 3.0.0

ags_message_delivery_find_sender ()

```
GList~*
ags_message_delivery_find_sender (AgsMessageDelivery *message_delivery,
                                  gchar *recipient_namespace,
                                  GObject *sender);
```

Find *sender* for *recipient_namespace* matching AgsMessageQueue. If *recipient_namespace* equals NULL match all available message queues.

Parameters

message_delivery	the AgsMessageDelivery	
recipient_namespace	the recipient namespace as string, maybe NULL	
sender	the sender as GObject	

Returns

all matching AgsMessageEnvelope as GList.

[element-type Ags.MessageEnvelope][transfer full]

Since: 3.0.0

ags_message_delivery_find_recipient ()

```
GList~*
ags_message_delivery_find_recipient (AgsMessageDelivery *message_delivery,
                                     gchar *recipient_namespace,
                                     GObject *recipient);
```

Find *recipient* for *recipient_namespace* matching AgsMessageQueue. If *recipient_namespace* equals NULL match all available message queues.

Parameters

message_delivery	the AgsMessageDelivery	
recipient_namespace	the recipient namespace as string, maybe NULL	
recipient	the recipient as GObject	

Returns

all matching AgsMessageEnvelope as GList.

[element-type Ags.MessageEnvelope][transfer full]

Since: 3.0.0

ags_message_delivery_query_message ()

```
GList~*
ags_message_delivery_query_message (AgsMessageDelivery *message_delivery,
                                    gchar *recipient_namespace,
                                    gchar *xpath);
```

Query *xpath* for *recipient_namespace* matching AgsMessageQueue. If *recipient_namespace* equals NULL match all available message queues.

Parameters

message_delivery	the AgsMessageDelivery	
recipient_namespace	the recipient namespace as string, maybe NULL	

xpath	the xpath to query

Returns

all matching AgsMessageEnvelope as GList.

[element-type Ags.MessageEnvelope][transfer full]

Since: 3.0.0

ags_message_delivery_get_instance ()

```
AgsMessageDelivery~*
ags_message_delivery_get_instance ();
```

Get the AgsMessageDelivery instance.

Returns

the AgsMessageDelivery.

[transfer none]

Since: 3.0.0

ags_message_delivery_new ()

```
AgsMessageDelivery~*
ags_message_delivery_new ();
```

Create a new AgsMessageDelivery.

Returns

the new AgsMessageDelivery

Since: 3.0.0

AGS_IS_MESSAGE_DELIVERY()

```
#define AGS_IS_MESSAGE_DELIVERY(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_MESSAGE_DELIVERY))
```

AGS_IS_MESSAGE_DELIVERY_CLASS()

```
#define AGS_IS_MESSAGE_DELIVERY_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_MESSAGE_DELIVERY))
```

AGS_MESSAGE_DELIVERY()

```
#define AGS_MESSAGE_DELIVERY(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MESSAGE_DELIVERY, AgsMessageDelivery))
```

AGS_MESSAGE_DELIVERY_CLASS()

```
#define AGS_MESSAGE_DELIVERY_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_MESSAGE_DELIVERY, AgsMessageDelivery))
```

AGS_MESSAGE_DELIVERY_GET_CLASS()

```
#define AGS_MESSAGE_DELIVERY_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_MESSAGE_DELIVERY, AgsMessageDeliveryClass))
```

ags_message_delivery_get_type ()

```
GType
ags_message_delivery_get_type ();
```

Types and Values

AGS_TYPE_MESSAGE_DELIVERY

```
#define AGS_TYPE_MESSAGE_DELIVERY                (ags_message_delivery_get_type())
```

struct AgsMessageDelivery

```
struct AgsMessageDelivery;
```

struct AgsMessageDeliveryClass

```
struct AgsMessageDeliveryClass {
  GObjectClass gobject;

};
```

Chapter 49

AgsMessageQueue

AgsMessageQueue — Message queue

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MESSAGE_QUEUE_GET_OBJ_MUTEX()
void	ags_message_queue_set_sender_namespace ()
gchar *	ags_message_queue_get_sender_namespace ()
void	ags_message_queue_set_recipient_namespace ()
gchar *	ags_message_queue_get_recipient_namespace ()
void	ags_message_queue_set_message_envelope ()
GList *	ags_message_queue_get_message_envelope ()
void	ags_message_queue_add_message_envelope ()
void	ags_message_queue_remove_message_envelope ()
GList *	ags_message_queue_find_sender ()
GList *	ags_message_queue_find_recipient ()
GList *	ags_message_queue_query_message ()
AgsMessageQueue *	ags_message_queue_new ()
#define	AGS_IS_MESSAGE_QUEUE()
#define	AGS_IS_MESSAGE_QUEUE_CLASS()
#define	AGS_MESSAGE_QUEUE()
#define	AGS_MESSAGE_QUEUE_CLASS()
#define	AGS_MESSAGE_QUEUE_GET_CLASS()
GType	ags_message_queue_get_type ()

Properties

char *	recipient-namespace	Read / Write
char *	sender-namespace	Read / Write

Types and Values

#define	AGS_TYPE_MESSAGE_QUEUE
struct	AgsMessageQueue
struct	AgsMessageQueueClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsMessageQueue
```

Includes

```
#include <ags/thread/ags_message_queue.h>
```

Description

The AgsMessageQueue acts as messages passing system. The queue hosts the AgsMessageEnvelope.

Functions

AGS_MESSAGE_QUEUE_GET_OBJ_MUTEX()

```
#define AGS_MESSAGE_QUEUE_GET_OBJ_MUTEX(obj) (&(((AgsMessageQueue *) obj)->obj_mutex))
```

ags_message_queue_set_sender_namespace ()

```
void
ags_message_queue_set_sender_namespace
                                (AgsMessageQueue *message_queue,
                                 gchar *sender_namespace);
```

Set *sender_namespace* of *message_queue* .

Parameters

message_queue	the AgsMessageQueue	
sender_namespace	the sender namespace	

Since: 3.6.16

ags_message_queue_get_sender_namespace ()

```
gchar~*
ags_message_queue_get_sender_namespace
                                (AgsMessageQueue *message_queue);
```

Get sender namespace of *message_queue* .

Parameters

message_queue	the AgsMessageQueue

Returns

the sender namespace.

[transfer full]

Since: 3.6.16

ags_message_queue_set_recipient_namespace ()

```
void
ags_message_queue_set_recipient_namespace
                                (AgsMessageQueue *message_queue,
                                 gchar *recipient_namespace);
```

Set *recipient_namespace* of *message_queue* .

Parameters

message_queue	the AgsMessageQueue
recipient_namespace	the recipient namespace

Since: 3.6.16

ags_message_queue_get_recipient_namespace ()

```
gchar~*
ags_message_queue_get_recipient_namespace
                                (AgsMessageQueue *message_queue);
```

Get recipient namespace of *message_queue* .

Parameters

message_queue	the AgsMessageQueue

Returns

the recipient namespace.

[transfer full]

Since: 3.6.16

ags_message_queue_set_message_envelope ()

```
void
ags_message_queue_set_message_envelope
                                (AgsMessageQueue *message_queue,
                                 GList *message_envelope);
```

Set *message_envelope* of *message_queue* .

Parameters

message_queue	the AgsMessageQueue	
message_envelope	the GList containing AgsMessageEnvelope.	*[element-type Ags.MessageEnvelope][transfer full]*

Since: 3.6.16

ags_message_queue_get_message_envelope ()

```
GList~*
ags_message_queue_get_message_envelope
                                (AgsMessageQueue *message_queue);
```

Get message envelope of *message_queue* .

Parameters

message_queue	the AgsMessageQueue	

Returns

the GList containing AgsMessageEnvelope.

[element-type Ags.MessageEnvelope][transfer none]

Since: 3.6.16

ags_message_queue_add_message_envelope ()

```
void
ags_message_queue_add_message_envelope
                                (AgsMessageQueue *message_queue,
                                 GObject *message_envelope);
```

Add *message* to *message_queue* .

Parameters

message_queue	the AgsMessageQueue	
message_envelope	the AgsMessageEnvelope	

Since: 3.0.0

ags_message_queue_remove_message_envelope ()

```
void
ags_message_queue_remove_message_envelope
                                (AgsMessageQueue *message_queue,
                                 GObject *message_envelope);
```

Remove *message* from *message_queue* .

Parameters

message_queue	the AgsMessageQueue	
message_envelope	the AgsMessageEnvelope	

Since: 3.0.0

ags_message_queue_find_sender ()

```
GList~*
ags_message_queue_find_sender (AgsMessageQueue *message_queue,
                               GObject *sender);
```

Find *sender* in *message_queue* 's messages.

Parameters

message_queue	the AgsMessageQueue	
sender	the sender as GObject	

Returns

all matching AgsMessageEnvelope as GList.

[element-type Ags.MessageEnvelope][transfer full]

Since: 3.0.0

ags_message_queue_find_recipient ()

```
GList~*
ags_message_queue_find_recipient (AgsMessageQueue *message_queue,
                                  GObject *recipient);
```

Find *recipient* in *message_queue* 's messages.

Parameters

message_queue	the AgsMessageQueue	
recipient	the recipient as GObject	

Returns

all matching AgsMessageEnvelope as GList.

[element-type Ags.MessageEnvelope][transfer full]

Since: 3.0.0

ags_message_queue_query_message ()

```
GList~*
ags_message_queue_query_message (AgsMessageQueue *message_queue,
                                 gchar *xpath);
```

Query *xpath* in *message_queue* 's messages.

Parameters

message_queue	the AgsMessageQueue	
xpath	the XPath as string	

Returns

all matching AgsMessageEnvelope as GList.

[element-type Ags.MessageEnvelope][transfer full]

Since: 3.0.0

ags_message_queue_new ()

```
AgsMessageQueue~*
ags_message_queue_new (gchar *sender_namespace);
```

Create a new instance of AgsMessageQueue.

Parameters

sender_namespace	the sender namespace	

Returns

the new AgsMessageQueue

Since: 3.0.0

AGS_IS_MESSAGE_QUEUE()

```
#define AGS_IS_MESSAGE_QUEUE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_MESSAGE_QUEUE))
```

AGS_IS_MESSAGE_QUEUE_CLASS()

```
#define AGS_IS_MESSAGE_QUEUE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_MESSAGE_QUEUE))
```

AGS_MESSAGE_QUEUE()

```
#define AGS_MESSAGE_QUEUE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MESSAGE_QUEUE, AgsMessageQueue))
```

AGS_MESSAGE_QUEUE_CLASS()

```
#define AGS_MESSAGE_QUEUE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_MESSAGE_QUEUE, AgsMessageQueue))
```

AGS_MESSAGE_QUEUE_GET_CLASS()

```
#define AGS_MESSAGE_QUEUE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_MESSAGE_QUEUE, AgsMessageQueueClass))
```

ags_message_queue_get_type ()

```
GType
ags_message_queue_get_type ();
```

Types and Values

AGS_TYPE_MESSAGE_QUEUE

```
#define AGS_TYPE_MESSAGE_QUEUE                 (ags_message_queue_get_type())
```

struct AgsMessageQueue

```
struct AgsMessageQueue;
```

struct AgsMessageQueueClass

```
struct AgsMessageQueueClass {
  GObjectClass gobject;

};
```

Property Details

The "recipient-namespace" property

```
"recipient-namespace"      char~*
```

The assigned namespace of recipient.

Owner: AgsMessageQueue

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "sender-namespace" property

```
"sender-namespace"         char~*
```

The assigned namespace of sender.

Owner: AgsMessageQueue

Flags: Read / Write

Default value: NULL

Since: 3.0.0

Chapter 50

AgsMessageEnvelope

AgsMessageEnvelope — message envelope

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_MESSAGE_ENVELOPE_GET_OBJ_MUTEX()
GObject *	ags_message_envelope_get_sender ()
GObject *	ags_message_envelope_get_recipient ()
xmlDoc *	ags_message_envelope_get_doc ()
void	ags_message_envelope_get_parameter ()
AgsMessageEnvelope *	ags_message_envelope_new ()
AgsMessageEnvelope *	ags_message_envelope_new_with_params ()
#define	AGS_IS_MESSAGE_ENVELOPE()
#define	AGS_IS_MESSAGE_ENVELOPE_CLASS()
#define	AGS_MESSAGE_ENVELOPE()
#define	AGS_MESSAGE_ENVELOPE_CLASS()
#define	AGS_MESSAGE_ENVELOPE_GET_CLASS()
GType	ags_message_envelope_get_type ()

Properties

GObject *	recipient	Read / Write
GObject *	sender	Read / Write
gpointer	xml-doc	Read / Write

Types and Values

#define	AGS_TYPE_MESSAGE_ENVELOPE
struct	AgsMessageEnvelope

struct	AgsMessageEnvelopeClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsMessageEnvelope
```

Includes

```
#include <ags/thread/ags_message_envelope.h>
```

Description

The AgsMessageEnvelope acts as messages passing system.

You create the message envelope with a xmlDoc which is your actual message, this is your `xml-doc` property.

You may provide additional parameters and obtain them by calling:

- ags_message_envelope_get_parameter()

Functions

AGS_MESSAGE_ENVELOPE_GET_OBJ_MUTEX()

```
#define AGS_MESSAGE_ENVELOPE_GET_OBJ_MUTEX(obj) (&(((AgsMessageEnvelope *) obj)->obj_mutex) ↩
    )
```

ags_message_envelope_get_sender ()

```
GObject~*
ags_message_envelope_get_sender (AgsMessageEnvelope *message_envelope);
```

Get sender.

Parameters

message_envelope	the AgsMessageEnvelope

Returns

the sender.

[transfer full]

Since: 3.0.0

ags_message_envelope_get_recipient ()

```
GObject~*
ags_message_envelope_get_recipient (AgsMessageEnvelope *message_envelope);
```

Get recipient.

Parameters

message_envelope	the AgsMessageEnvelope

Returns

the recipient.

[transfer full]

Since: 3.0.0

ags_message_envelope_get_doc ()

```
xmlDoc~*
ags_message_envelope_get_doc (AgsMessageEnvelope *message_envelope);
```

Get xmlDoc.

Parameters

message_envelope	the AgsMessageEnvelope

Returns

the doc.

[transfer none]

Since: 3.0.0

ags_message_envelope_get_parameter ()

```
void
ags_message_envelope_get_parameter (AgsMessageEnvelope *message_envelope,
                                    guint *n_params,
                                    gchar ***parameter_name,
                                    GValue **value);
```

Get parameters.

Parameters

message_envelope	the AgsMessageEnvelope	
n_params	return location of parameter count	
parameter_name	return location NULL terminated string vector containing parameter name	
value	return location of GValue array containing values	

Since: 3.0.0

ags_message_envelope_new ()

```
AgsMessageEnvelope~*
ags_message_envelope_new (GObject *sender,
                          GObject *recipient,
                          xmlDoc *doc);
```

Create a new instance of AgsMessageEnvelope.

Parameters

sender	the GObject as sender	
recipient	the GObject as recipient	
doc	the xmlDoc	

Returns

the new AgsMessageEnvelope

Since: 3.0.0

ags_message_envelope_new_with_params ()

```
AgsMessageEnvelope~*
ags_message_envelope_new_with_params (GObject *sender,
                                      GObject *recipient,
                                      xmlDoc *doc,
                                      guint n_params,
                                      gchar **parameter_name,
                                      GValue *value);
```

Create a new instance of AgsMessageEnvelope.

Parameters

sender	the GObject as sender	
recipient	the GObject as recipient	
doc	the xmlDoc	

n_params	n params	
parameter_name	NULL terminated string vector containing parameter names	
value	the GValue array containing values	

Returns

the new AgsMessageEnvelope

Since: 3.0.0

AGS_IS_MESSAGE_ENVELOPE()

```
#define AGS_IS_MESSAGE_ENVELOPE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_MESSAGE_ENVELOPE))
```

AGS_IS_MESSAGE_ENVELOPE_CLASS()

```
#define AGS_IS_MESSAGE_ENVELOPE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_MESSAGE_ENVELOPE))
```

AGS_MESSAGE_ENVELOPE()

```
#define AGS_MESSAGE_ENVELOPE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_MESSAGE_ENVELOPE, AgsMessageEnvelope))
```

AGS_MESSAGE_ENVELOPE_CLASS()

```
#define AGS_MESSAGE_ENVELOPE_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_MESSAGE_ENVELOPE, AgsMessageEnvelope))
```

AGS_MESSAGE_ENVELOPE_GET_CLASS()

```
#define AGS_MESSAGE_ENVELOPE_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_MESSAGE_ENVELOPE, AgsMessageEnvelopeClass))
```

ags_message_envelope_get_type ()

```
GType
ags_message_envelope_get_type ();
```

Types and Values

AGS_TYPE_MESSAGE_ENVELOPE

```
#define AGS_TYPE_MESSAGE_ENVELOPE                 (ags_message_envelope_get_type())
```

struct AgsMessageEnvelope

```
struct AgsMessageEnvelope;
```

struct AgsMessageEnvelopeClass

```
struct AgsMessageEnvelopeClass {
  GObjectClass gobject;

};
```

Property Details

The "recipient" property

```
  "recipient"                 GObject~*
```

The assigned recipient.

Owner: AgsMessageEnvelope

Flags: Read / Write

Since: 3.0.0

The "sender" property

```
  "sender"                    GObject~*
```

The assigned sender.

Owner: AgsMessageEnvelope

Flags: Read / Write

Since: 3.0.0

The "xml-doc" property

```
  "xml-doc"                   gpointer
```

The assigned xml-doc.

Owner: AgsMessageEnvelope

Flags: Read / Write

Since: 3.0.0

Chapter 51

AgsReturnableThread

AgsReturnableThread — returnable thread

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_RETURNABLE_THREAD_GET_RESET_MUTEX()
void	(*AgsReturnableThreadCallback) ()
gboolean	ags_returnable_thread_test_flags ()
void	ags_returnable_thread_set_flags ()
void	ags_returnable_thread_unset_flags ()
void	ags_returnable_thread_safe_run ()
void	ags_returnable_thread_connect_safe_run ()
void	ags_returnable_thread_disconnect_safe_run ()
AgsReturnableThread *	ags_returnable_thread_new ()
#define	AGS_IS_RETURNABLE_THREAD()
#define	AGS_IS_RETURNABLE_THREAD_CLASS()
#define	AGS_RETURNABLE_THREAD()
#define	AGS_RETURNABLE_THREAD_CLASS()
#define	AGS_RETURNABLE_THREAD_GET_CLASS()
GType	ags_returnable_thread_flags_get_type ()
GType	ags_returnable_thread_get_type ()

Properties

GObject *	thread-pool	Read / Write

Signals

void	safe-run	Run Last

Types and Values

#define	AGS_RETURNABLE_THREAD_DEFAULT_JIFFIE
enum	AgsReturnableThreadFlags
#define	AGS_TYPE_RETURNABLE_THREAD
#define	AGS_TYPE_RETURNABLE_THREAD_FLAGS
struct	AgsReturnableThread
struct	AgsReturnableThreadClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsReturnableThreadFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsThread
        &#x2570;&#x2500;&#x2500; AgsReturnableThread
```

Implemented Interfaces

AgsReturnableThread implements AgsConnectable.

Includes

```
#include <ags/thread/ags_returnable_thread.h>
```

Description

The AgsReturnableThread acts as thread. It should return after a short while because of limited thread pool.

Functions

AGS_RETURNABLE_THREAD_GET_RESET_MUTEX()

```
#define AGS_RETURNABLE_THREAD_GET_RESET_MUTEX(obj) (&(((AgsReturnableThread *) obj)-> ↩
    reset_mutex))
```

AgsReturnableThreadCallback ()

```
void
(*AgsReturnableThreadCallback) (AgsReturnableThread *returnable_thread,
                                gpointer data);
```

ags_returnable_thread_test_flags ()

```
gboolean
ags_returnable_thread_test_flags (AgsReturnableThread *returnable_thread,
                                  guint flags);
```

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

Parameters

returnable_thread	the AgsReturnableThread	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_returnable_thread_set_flags ()

```
void
ags_returnable_thread_set_flags (AgsReturnableThread *returnable_thread,
                                 guint flags);
```

Set flags.

Parameters

returnable_thread	the AgsReturnableThread	
flags	the flags	

Since: 3.0.0

ags_returnable_thread_unset_flags ()

```
void
ags_returnable_thread_unset_flags (AgsReturnableThread *returnable_thread,
                                   guint flags);
```

Unset flags.

Parameters

returnable_thread	the AgsReturnableThread	
flags	the flags	

Since: 3.0.0

ags_returnable_thread_safe_run ()

```
void
ags_returnable_thread_safe_run (AgsReturnableThread *returnable_thread);
```

ags_returnable_thread_connect_safe_run ()

```
void
ags_returnable_thread_connect_safe_run
                                (AgsReturnableThread *returnable_thread,
                                 AgsReturnableThreadCallback callback);
```

Connects *callback* to *thread* .

Parameters

returnable_thread	the thread to connect	
callback	the callback.	*[scope call]*

Since: 3.0.0

ags_returnable_thread_disconnect_safe_run ()

```
void
ags_returnable_thread_disconnect_safe_run
                                (AgsReturnableThread *returnable_thread);
```

Disconnects callback of *thread* .

Parameters

returnable_thread	the thread to disconnect	

Since: 3.0.0

ags_returnable_thread_new ()

```
AgsReturnableThread~*
ags_returnable_thread_new (GObject *thread_pool);
```

Create a new instance of AgsReturnableThread.

Parameters

thread_pool	the AgsThreadPool	

Returns

the new AgsReturnableThread

Since: 3.0.0

AGS_IS_RETURNABLE_THREAD()

```
#define AGS_IS_RETURNABLE_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_RETURNABLE_THREAD))
```

AGS_IS_RETURNABLE_THREAD_CLASS()

```
#define AGS_IS_RETURNABLE_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_RETURNABLE_THREAD))
```

AGS_RETURNABLE_THREAD()

```
#define AGS_RETURNABLE_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_RETURNABLE_THREAD, AgsReturnableThread))
```

AGS_RETURNABLE_THREAD_CLASS()

```
#define AGS_RETURNABLE_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_RETURNABLE_THREAD, AgsReturnableThreadClass))
```

AGS_RETURNABLE_THREAD_GET_CLASS()

```
#define AGS_RETURNABLE_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_RETURNABLE_THREAD, AgsReturnableThreadClass))
```

ags_returnable_thread_flags_get_type ()

```
GType
ags_returnable_thread_flags_get_type ();
```

ags_returnable_thread_get_type ()

```
GType
ags_returnable_thread_get_type ();
```

Types and Values

AGS_RETURNABLE_THREAD_DEFAULT_JIFFIE

```
#define AGS_RETURNABLE_THREAD_DEFAULT_JIFFIE (AGS_THREAD_DEFAULT_MAX_PRECISION)
```

enum AgsReturnableThreadFlags

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

Members

AGS_RETURNABLE_THREAD_IN_USE	the thread is in use
AGS_RETURNABLE_THREAD_RESET	not used
AGS_RETURNABLE_THREAD_RUN_ONCE	call AgsThread::run() only one time

AGS_TYPE_RETURNABLE_THREAD

```
#define AGS_TYPE_RETURNABLE_THREAD                (ags_returnable_thread_get_type())
```

AGS_TYPE_RETURNABLE_THREAD_FLAGS

```
#define AGS_TYPE_RETURNABLE_THREAD_FLAGS          (ags_returnable_thread_flags_get_type())
```

struct AgsReturnableThread

```
struct AgsReturnableThread;
```

struct AgsReturnableThreadClass

```
struct AgsReturnableThreadClass {
  AgsThreadClass thread;

  void (*safe_run)(AgsReturnableThread *returnable_thread);
};
```

Property Details

The "thread-pool" property

```
  "thread-pool"              GObject~*
```

The assigned AgsThreadPool providing default settings.

Owner: AgsReturnableThread

Flags: Read / Write

Since: 3.0.0

Signal Details

The "safe-run" signal

```
void
user_function (AgsReturnableThread *returnable_thread,
               gpointer             user_data)
```

The ::safe-run is invoked durin AgsThread::run as a context safe wrapper.

Parameters

returnable_thread	the AgsReturnableThread	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 52

AgsTaskLauncher

AgsTaskLauncher — Task launcher thread-safe context within tree

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TASK_LAUNCHER_GET_OBJ_MUTEX()
gboolean	ags_task_launcher_test_flags ()
void	ags_task_launcher_set_flags ()
void	ags_task_launcher_unset_flags ()
void	ags_task_launcher_attach ()
void	ags_task_launcher_add_task ()
void	ags_task_launcher_add_task_all ()
void	ags_task_launcher_add_cyclic_task ()
void	ags_task_launcher_remove_cyclic_task ()
void	ags_task_launcher_run ()
void	ags_task_launcher_sync_run ()
AgsTaskLauncher *	ags_task_launcher_new ()
#define	AGS_IS_TASK_LAUNCHER()
#define	AGS_IS_TASK_LAUNCHER_CLASS()
#define	AGS_TASK_LAUNCHER()
#define	AGS_TASK_LAUNCHER_CLASS()
#define	AGS_TASK_LAUNCHER_GET_CLASS()
GType	ags_task_launcher_get_type ()

Properties

gpointer	cyclic-task	Read / Write
gpointer	task	Read / Write

Signals

void	run	Run Last

Types and Values

#define	AGS_TYPE_TASK_LAUNCHER_FLAGS
#define	AGS_TYPE_TASK_LAUNCHER
struct	AgsTaskLauncher
struct	AgsTaskLauncherClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTaskLauncher
```

Implemented Interfaces

AgsTaskLauncher implements AgsConnectable.

Includes

```
#include <ags/thread/ags_task_launcher.h>
```

Description

The AgsTaskLauncher acts as task launcher. The task launcher provides you a thread-safe context within tree tree.

It might be used to inject work from one thread to another, too.

AgsTaskLauncher::run() is called from AgsThread::clock() default implementation.

There are some threads providing injection via dedicated AgsTaskLauncher.

Functions

AGS_TASK_LAUNCHER_GET_OBJ_MUTEX()

```
#define AGS_TASK_LAUNCHER_GET_OBJ_MUTEX(obj) (&(((AgsTaskLauncher *) obj)->obj_mutex))
```

ags_task_launcher_test_flags ()

```
gboolean
ags_task_launcher_test_flags (AgsTaskLauncher *task_launcher,
                              guint flags);
```

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

Parameters

task_launcher	the AgsTaskLauncher	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_task_launcher_set_flags ()

```
void
ags_task_launcher_set_flags (AgsTaskLauncher *task_launcher,
                             guint flags);
```

Enable a feature of AgsTaskLauncher.

Parameters

task_launcher	the AgsTaskLauncher	
flags	see enum AgsTaskLauncherFlags	

Since: 3.0.0

ags_task_launcher_unset_flags ()

```
void
ags_task_launcher_unset_flags (AgsTaskLauncher *task_launcher,
                               guint flags);
```

Disable a feature of AgsTaskLauncher.

Parameters

task_launcher	the AgsTaskLauncher	
flags	see enum AgsTaskLauncherFlags	

Since: 3.0.0

ags_task_launcher_attach ()

```
void
ags_task_launcher_attach (AgsTaskLauncher *task_launcher,
                          GMainContext *main_context);
```

Attach *task_launcher* to *main_context* .

Parameters

task_launcher	the AgsTaskLauncher	
main_context	the GMainContext	

Since: 3.0.0

ags_task_launcher_add_task ()

```
void
ags_task_launcher_add_task (AgsTaskLauncher *task_launcher,
                            AgsTask *task);
```

Add *task* to *task_launcher* .

Parameters

task_launcher	the AgsTaskLauncher	
task	the AgsTask	

Since: 3.0.0

ags_task_launcher_add_task_all ()

```
void
ags_task_launcher_add_task_all (AgsTaskLauncher *task_launcher,
                                GList *list);
```

Add all *list* to *task_launcher* .

Parameters

task_launcher	the AgsTaskLauncher	
list	the GList containing AgsTask.	*[element-type Ags.Task][transfer none]*

Since: 3.0.0

ags_task_launcher_add_cyclic_task ()

```
void
ags_task_launcher_add_cyclic_task (AgsTaskLauncher *task_launcher,
                                   AgsTask *cyclic_task);
```

Add *cyclic_task* to *task_launcher* .

Parameters

task_launcher	the AgsTaskLauncher	
cyclic_task	the cyclic AgsTask	

Since: 3.0.0

ags_task_launcher_remove_cyclic_task ()

```
void
ags_task_launcher_remove_cyclic_task (AgsTaskLauncher *task_launcher,
                                      AgsTask *cyclic_task);
```

Remove *cyclic_task* from *task_launcher* .

Parameters

task_launcher	the AgsTaskLauncher	
cyclic_task	the cyclic AgsTask	

Since: 3.0.0

ags_task_launcher_run ()

```
void
ags_task_launcher_run (AgsTaskLauncher *task_launcher);
```

Run tasks.

Parameters

task_launcher	the AgsTaskLauncher	

Since: 3.0.0

ags_task_launcher_sync_run ()

```
void
ags_task_launcher_sync_run (AgsTaskLauncher *task_launcher);
```

Sync run tasks.

Parameters

task_launcher	the AgsTaskLauncher	

Since: 3.0.0

ags_task_launcher_new ()

```
AgsTaskLauncher~*
ags_task_launcher_new ();
```

Create a new AgsTaskLauncher.

Returns

the new AgsTaskLauncher

Since: 3.0.0

AGS_IS_TASK_LAUNCHER()

```
#define AGS_IS_TASK_LAUNCHER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_TASK_LAUNCHER))
```

AGS_IS_TASK_LAUNCHER_CLASS()

```
#define AGS_IS_TASK_LAUNCHER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_TASK_LAUNCHER))
```

AGS_TASK_LAUNCHER()

```
#define AGS_TASK_LAUNCHER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TASK_LAUNCHER, AgsTaskLauncher))
```

AGS_TASK_LAUNCHER_CLASS()

```
#define AGS_TASK_LAUNCHER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_TASK_LAUNCHER, AgsTaskLauncher))
```

AGS_TASK_LAUNCHER_GET_CLASS()

```
#define AGS_TASK_LAUNCHER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_TASK_LAUNCHER, AgsTaskLauncherClass))
```

ags_task_launcher_get_type ()

```
GType
ags_task_launcher_get_type ();
```

Types and Values

AGS_TYPE_TASK_LAUNCHER_FLAGS

```
#define AGS_TYPE_TASK_LAUNCHER_FLAGS           (ags_task_launcher_flags_get_type())
```

AGS_TYPE_TASK_LAUNCHER

```
#define AGS_TYPE_TASK_LAUNCHER                (ags_task_launcher_get_type())
```

struct AgsTaskLauncher

```
struct AgsTaskLauncher;
```

struct AgsTaskLauncherClass

```
struct AgsTaskLauncherClass {
  GObjectClass gobject;

  void (*run)(AgsTaskLauncher *task_launcher);
};
```

Property Details

The "cyclic-task" property

```
  "cyclic-task"              gpointer
```

The assigned cyclic AgsTask to launch.

[transfer full]

Owner: AgsTaskLauncher

Flags: Read / Write

Since: 3.0.0

The "task" property

```
  "task"                     gpointer
```

The assigned AgsTask to launch.

[transfer full]

Owner: AgsTaskLauncher

Flags: Read / Write

Since: 3.0.0

Signal Details

The "run" signal

```
void
user_function (AgsTaskLauncher *task_launcher,
               gpointer         user_data)
```

The ::run signal is invoked to run "task"

Parameters

task_launcher	the AgsTaskLauncher	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 53

AgsTask

AgsTask — Thread-safe task

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TASK_GET_OBJ_MUTEX()
gboolean	ags_task_test_flags ()
void	ags_task_set_flags ()
void	ags_task_unset_flags ()
void	ags_task_launch ()
void	ags_task_failure ()
AgsTask *	ags_task_new ()
#define	AGS_IS_TASK()
#define	AGS_IS_TASK_CLASS()
#define	AGS_TASK()
#define	AGS_TASK_CLASS()
#define	AGS_TASK_GET_CLASS()
GType	ags_task_get_type ()

Properties

AgsTaskLauncher *	task-launcher	Read / Write

Signals

void	failure	Run Last
void	launch	Run Last

Types and Values

#define	AGS_TYPE_TASK_FLAGS
enum	AgsTaskFlags
#define	AGS_TYPE_TASK
struct	AgsTask
struct	AgsTaskClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsTask
```

Includes

```
#include <ags/thread/ags_task.h>
```

Description

AgsTask object acts an interceptor in a thread safe context. It's AgsTask::launch() function is normally called by AgsTaskLauncher.

The AgsTaskLauncher within thread tree provides thread-safety within entire tree context. You obtain the task launcher of the thread tree by calling AgsConcurrencyProvider::get_task_launcher()

There are some task launcher with threads, too. In order to inject from one thread to another.

Functions

AGS_TASK_GET_OBJ_MUTEX()

```
#define AGS_TASK_GET_OBJ_MUTEX(obj) (&(((AgsTask *) obj)->obj_mutex))
```

ags_task_test_flags ()

```
gboolean
ags_task_test_flags (AgsTask *task,
                     guint flags);
```

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

Parameters

task	the AgsTask	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_task_set_flags ()

```
void
ags_task_set_flags (AgsTask *task,
                    guint flags);
```

Enable a feature of AgsTask.

Parameters

task	the AgsTask	
flags	see enum AgsTaskFlags	

Since: 3.0.0

ags_task_unset_flags ()

```
void
ags_task_unset_flags (AgsTask *task,
                      guint flags);
```

Disable a feature of AgsTask.

Parameters

task	the AgsTask	
flags	see enum AgsTaskFlags	

Since: 3.0.0

ags_task_launch ()

```
void
ags_task_launch (AgsTask *task);
```

Intercept task.

Parameters

task	an AgsTask	

Since: 3.0.0

ags_task_failure ()

```
void
ags_task_failure (AgsTask *task,
                  GError *error);
```

Signals failure of task.

Parameters

task	an AgsTask	
error	is NULL on success	

Since: 3.0.0

ags_task_new ()

```
AgsTask~*
ags_task_new ();
```

Create a new AgsTask.

Returns

the new AgsTask

Since: 3.0.0

AGS_IS_TASK()

```
#define AGS_IS_TASK(obj)              (G_TYPE_CHECK_INSTANCE_TYPE((obj), AGS_TYPE_TASK))
```

AGS_IS_TASK_CLASS()

```
#define AGS_IS_TASK_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE((class), AGS_TYPE_TASK))
```

AGS_TASK()

```
#define AGS_TASK(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_TASK,  ↩
    AgsTask))
```

AGS_TASK_CLASS()

```
#define AGS_TASK_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class), AGS_TYPE_TASK,  ↩
    AgsTaskClass))
```

AGS_TASK_GET_CLASS()

```
#define AGS_TASK_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS((obj), AGS_TYPE_TASK,  ↩
    AgsTaskClass))
```

ags_task_get_type ()

```
GType
ags_task_get_type ();
```

Types and Values

AGS_TYPE_TASK_FLAGS

```
#define AGS_TYPE_TASK_FLAGS                 (ags_task_flags_get_type())
```

enum AgsTaskFlags

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

Members

AGS_TASK_LOCKED	the task is locked
AGS_TASK_CYCLIC	call task repeatedly

AGS_TYPE_TASK

```
#define AGS_TYPE_TASK                 (ags_task_get_type())
```

struct AgsTask

```
struct AgsTask;
```

struct AgsTaskClass

```
struct AgsTaskClass {
  GObjectClass gobject;

  void (*launch)(AgsTask *task);

  void (*failure)(AgsTask *task, GError *error);
};
```

Property Details

The "task-launcher" property

```
  "task-launcher"            AgsTaskLauncher~*
```

The assigned AgsTaskLauncher

Owner: AgsTask

Flags: Read / Write

Since: 3.0.0

Signal Details

The "failure" signal

```
void
user_function (AgsTask *task,
               gpointer error,
               gpointer user_data)
```

The ::failure signal is emited if ::launch fails

Parameters

task	the AgsTask failed to do its work.	
error	the GError	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "launch" signal

```
void
user_function (AgsTask *task,
               gpointer user_data)
```

The ::launch signal is emited in a thread safe context

Parameters

task	the AgsTask to launch	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 54

AgsTaskCompletion

AgsTaskCompletion — task completion

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TASK_COMPLETION_GET_OBJ_MUTEX()
gboolean	ags_task_completion_test_flags ()
void	ags_task_completion_set_flags ()
void	ags_task_completion_unset_flags ()
void	ags_task_completion_complete ()
AgsTaskCompletion *	ags_task_completion_new ()
#define	AGS_IS_TASK_COMPLETION()
#define	AGS_IS_TASK_COMPLETION_CLASS()
#define	AGS_TASK_COMPLETION()
#define	AGS_TASK_COMPLETION_CLASS()
#define	AGS_TASK_COMPLETION_GET_CLASS()
GType	ags_task_completion_flags_get_type ()
GType	ags_task_completion_get_type ()

Properties

GObject *	task	Read / Write

Signals

void	complete	Run Last

Types and Values

enum	AgsTaskCompletionFlags
#define	AGS_TYPE_TASK_COMPLETION
#define	AGS_TYPE_TASK_COMPLETION_FLAGS
struct	AgsTaskCompletion
struct	AgsTaskCompletionClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsTaskCompletionFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsTaskCompletion
```

Implemented Interfaces

AgsTaskCompletion implements AgsConnectable.

Includes

```
#include <ags/thread/ags_task_completion.h>
```

Description

The AgsTaskCompletion should be used to complete tasks from caller thread.

Functions

AGS_TASK_COMPLETION_GET_OBJ_MUTEX()

```
#define AGS_TASK_COMPLETION_GET_OBJ_MUTEX(obj) (&(((AgsTaskCompletion *) obj)->obj_mutex))
```

ags_task_completion_test_flags ()

```
gboolean
ags_task_completion_test_flags (AgsTaskCompletion *task_completion,
                                guint flags);
```

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

Parameters

task_completion	the AgsTaskCompletion	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_task_completion_set_flags ()

```
void
ags_task_completion_set_flags (AgsTaskCompletion *task_completion,
                               guint flags);
```

Enable a feature of AgsTaskCompletion.

Parameters

task_completion	the AgsTaskCompletion	
flags	see enum AgsTaskCompletionFlags	

Since: 3.0.0

ags_task_completion_unset_flags ()

```
void
ags_task_completion_unset_flags (AgsTaskCompletion *task_completion,
                                 guint flags);
```

Disable a feature of AgsTaskCompletion.

Parameters

task_completion	the AgsTaskCompletion	
flags	see enum AgsTaskCompletionFlags	

Since: 3.0.0

ags_task_completion_complete ()

```
void
ags_task_completion_complete (AgsTaskCompletion *task_completion);
```

Emit ::complete signal

Parameters

task_completion	the AgsTaskCompletion	

Since: 3.0.0

ags_task_completion_new ()

```
AgsTaskCompletion~*
ags_task_completion_new (GObject *task,
                         gpointer data);
```

Create a new instance of AgsTaskCompletion.

Parameters

task	the AgsTask	
data	pointer to user data	

Returns

the new AgsTaskCompletion

Since: 3.0.0

AGS_IS_TASK_COMPLETION()

```
#define AGS_IS_TASK_COMPLETION(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_TASK_COMPLETION))
```

AGS_IS_TASK_COMPLETION_CLASS()

```
#define AGS_IS_TASK_COMPLETION_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_TASK_COMPLETION))
```

AGS_TASK_COMPLETION()

```
#define AGS_TASK_COMPLETION(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TASK_COMPLETION, AgsTaskCompletion))
```

AGS_TASK_COMPLETION_CLASS()

```
#define AGS_TASK_COMPLETION_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_TASK_COMPLETION, AgsTaskCompletionClass))
```

AGS_TASK_COMPLETION_GET_CLASS()

```
#define AGS_TASK_COMPLETION_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_TASK_COMPLETION, AgsTaskCompletionClass))
```

ags_task_completion_flags_get_type ()

```
GType
ags_task_completion_flags_get_type ();
```

ags_task_completion_get_type ()

```
GType
ags_task_completion_get_type ();
```

Types and Values

enum AgsTaskCompletionFlags

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

Members

AGS_TASK_COMPLETION_QUEUED	the as-signed task has been queued
AGS_TASK_COMPLETION_BUSY	the as-signed task is busy
AGS_TASK_COMPLETION_READY	the as-signed task is ready
AGS_TASK_COMPLETION_COMPLETED	the as-signed task has been com-pleted

AGS_TYPE_TASK_COMPLETION

```
#define AGS_TYPE_TASK_COMPLETION                (ags_task_completion_get_type())
```

AGS_TYPE_TASK_COMPLETION_FLAGS

```
#define AGS_TYPE_TASK_COMPLETION_FLAGS          (ags_task_completion_flags_get_type())
```

struct AgsTaskCompletion

```
struct AgsTaskCompletion;
```

struct AgsTaskCompletionClass

```
struct AgsTaskCompletionClass {
  GObjectClass gobject;

  void (*complete)(AgsTaskCompletion *task_completion);
};
```

Property Details

The "task" property

```
  "task"                      GObject~*
```

The assigned task.

Owner: AgsTaskCompletion

Flags: Read / Write

Since: 3.0.0

Signal Details

The "complete" signal

```
void
user_function (AgsTaskCompletion *task_completion,
               gpointer           user_data)
```

The ::complete signal is emited as a task was finished

Parameters

task_completion	the object to complete.	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 55

AgsThread

AgsThread — Thread class with base frequency

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_THREAD_GET_OBJ_MUTEX()
#define	AGS_THREAD_GET_WAIT_MUTEX()
#define	AGS_THREAD_GET_WAIT_COND()
#define	AGS_THREAD_GET_TIC_MUTEX()
#define	AGS_THREAD_GET_TIC_COND()
#define	AGS_THREAD_GET_START_MUTEX()
#define	AGS_THREAD_GET_START_COND()
gboolean	ags_thread_global_get_use_sync_counter ()
gboolean	ags_thread_test_flags ()
void	ags_thread_set_flags ()
void	ags_thread_unset_flags ()
gboolean	ags_thread_test_status_flags ()
void	ags_thread_set_status_flags ()
void	ags_thread_unset_status_flags ()
void	ags_thread_clear_status_flags ()
gboolean	ags_thread_test_sync_tic_flags ()
void	ags_thread_set_sync_tic_flags ()
void	ags_thread_unset_sync_tic_flags ()
void	ags_thread_clear_sync_tic_flags ()
void	ags_thread_set_current_sync_tic ()
guint	ags_thread_get_current_sync_tic ()
void	ags_thread_set_delay ()
gdouble	ags_thread_get_delay ()
void	ags_thread_set_frequency ()
gdouble	ags_thread_get_frequency ()
void	ags_thread_set_max_precision ()
gdouble	ags_thread_get_max_precision ()
AgsThread *	ags_thread_find_type ()

AgsThread *	ags_thread_self ()
AgsThread *	ags_thread_parent ()
AgsThread *	ags_thread_next ()
AgsThread *	ags_thread_prev ()
AgsThread *	ags_thread_children ()
AgsThread *	ags_thread_get_toplevel ()
AgsThread *	ags_thread_first ()
AgsThread *	ags_thread_last ()
void	ags_thread_lock ()
gboolean	ags_thread_trylock ()
void	ags_thread_unlock ()
void	ags_thread_remove_child ()
void	ags_thread_add_child ()
void	ags_thread_add_child_extended ()
gboolean	ags_thread_is_current_ready ()
gboolean	ags_thread_is_tree_ready_recursive ()
void	ags_thread_prepare_current_sync ()
void	ags_thread_prepare_tree_sync_recursive ()
void	ags_thread_set_current_sync ()
void	ags_thread_set_tree_sync_recursive ()
guint	ags_thread_clock ()
void	ags_thread_add_start_queue ()
void	ags_thread_add_start_queue_all ()
void	ags_thread_start ()
void	ags_thread_run ()
void	ags_thread_stop ()
AgsThread *	ags_thread_new ()
#define	AGS_IS_THREAD()
#define	AGS_IS_THREAD_CLASS()
#define	AGS_THREAD()
#define	AGS_THREAD_CLASS()
#define	AGS_THREAD_GET_CLASS()
GType	ags_thread_flags_get_type ()
GType	ags_thread_get_type ()
GType	ags_thread_status_flags_get_type ()
GType	ags_thread_sync_tic_flags_get_type ()

Properties

double	delay	Read / Write
double	frequency	Read / Write
double	max-precision	Read / Write

Signals

guint	clock	Run Last
void	run	Run Last
void	start	Run Last
void	stop	Run Last

Types and Values

#define	AGS_THREAD_HERTZ_JIFFIE
#define	AGS_THREAD_YIELD_JIFFIE
#define	AGS_THREAD_DEFAULT_JIFFIE
#define	AGS_THREAD_DEFAULT_MAX_PRECISION
#define	AGS_THREAD_MAX_PRECISION
#define	AGS_THREAD_DEFAULT_ATTACK
#define	AGS_THREAD_TOLERANCE
enum	AgsThreadFlags
enum	AgsThreadStatusFlags
enum	AgsThreadSyncTicFlags
#define	AGS_TYPE_THREAD
#define	AGS_TYPE_THREAD_FLAGS
#define	AGS_TYPE_THREAD_STATUS_FLAGS
#define	AGS_TYPE_THREAD_SYNC_TIC_FLAGS
struct	AgsThread
struct	AgsThreadClass

Object Hierarchy

```
GFlags
&#x251c;&#x2500;&#x2500; AgsThreadFlags
&#x251c;&#x2500;&#x2500; AgsThreadStatusFlags
&#x2570;&#x2500;&#x2500; AgsThreadSyncTicFlags
GObject
&#x2570;&#x2500;&#x2500; AgsThread
    &#x251c;&#x2500;&#x2500; AgsWorkerThread
    &#x251c;&#x2500;&#x2500; AgsGenericMainLoop
    &#x2570;&#x2500;&#x2500; AgsReturnableThread
```

Implemented Interfaces

AgsThread implements AgsConnectable.

Includes

```
#include <ags/thread/ags_thread.h>
```

Description

The AgsThread base class. It supports organizing them within a tree, perform syncing and frequencies.

The thread class synchronizes within tree at `max-precision` property JIFFIE. AgsThread::run() is called at `frequency` property JIFFIE.

The AgsThread::run() function can sync to the very same tic with AGS_THREAD_IMMEDIATE_SYNC flag, sync to the previous or next tic with AGS_THREAD_INTERMEDIATE_PRE_SYNC or AGS_THREAD_INTERMEDIATE_POST_SYNC flag set.

The AgsTread::clock() function is called at `max-precision` property rate per second.

Note you usually have a default source giving you timing hints i.e. poll is used. If not you can still set the AGS_THREAD_TIME_ACCOU
flags, causing a sleep for every tic.

Further there is no time promise within one second. But you should make sure you hit the real-time dead-line or use a worker thread instead for time expensive work or tasks.

ags_thread_self() returns you your own AgsThread you are running in or NULL if not applicable.

Functions

AGS_THREAD_GET_OBJ_MUTEX()

```
#define AGS_THREAD_GET_OBJ_MUTEX(obj) (&(((AgsThread *) obj)->obj_mutex))
```

AGS_THREAD_GET_WAIT_MUTEX()

```
#define AGS_THREAD_GET_WAIT_MUTEX(obj) (&(((AgsThread *) obj)->wait_mutex))
```

AGS_THREAD_GET_WAIT_COND()

```
#define AGS_THREAD_GET_WAIT_COND(obj) (&(((AgsThread *) obj)->wait_cond))
```

AGS_THREAD_GET_TIC_MUTEX()

```
#define AGS_THREAD_GET_TIC_MUTEX(obj) (&(((AgsThread *) obj)->tic_mutex))
```

AGS_THREAD_GET_TIC_COND()

```
#define AGS_THREAD_GET_TIC_COND(obj) (&(((AgsThread *) obj)->tic_cond))
```

AGS_THREAD_GET_START_MUTEX()

```
#define AGS_THREAD_GET_START_MUTEX(obj) (&(((AgsThread *) obj)->start_mutex))
```

AGS_THREAD_GET_START_COND()

```
#define AGS_THREAD_GET_START_COND(obj) (&(((AgsThread *) obj)->start_cond))
```

ags_thread_global_get_use_sync_counter ()

```
gboolean
ags_thread_global_get_use_sync_counter
                                ();
```

Get global config value use sync counter.

Returns

if TRUE use sync counter, else not

Since: 3.0.0

ags_thread_test_flags ()

```
gboolean
ags_thread_test_flags (AgsThread *thread,
                       guint flags);
```

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

Parameters

thread	the AgsThread	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_thread_set_flags ()

```
void
ags_thread_set_flags (AgsThread *thread,
                      guint flags);
```

Set flags.

Parameters

thread	the AgsThread	
flags	the flags	

Since: 3.0.0

ags_thread_unset_flags ()

```
void
ags_thread_unset_flags (AgsThread *thread,
                        guint flags);
```

Unset flags.

Parameters

thread	the AgsThread	
flags	the flags	

Since: 3.0.0

ags_thread_test_status_flags ()

```
gboolean
ags_thread_test_status_flags (AgsThread *thread,
                              guint status_flags);
```

Test *status_flags* to be set on *thread* .

Parameters

thread	the AgsThread	
status_flags	the status flags	

Returns

TRUE if status flags are set, else FALSE

Since: 3.0.0

ags_thread_set_status_flags ()

```
void
ags_thread_set_status_flags (AgsThread *thread,
                             guint status_flags);
```

Set status flags.

Parameters

thread	the AgsThread	
status_flags	the status flags	

Since: 3.0.0

ags_thread_unset_status_flags ()

```
void
ags_thread_unset_status_flags (AgsThread *thread,
                               guint status_flags);
```

Unset status flags.

Parameters

thread	the AgsThread	
status_flags	the status flags	

Since: 3.0.0

ags_thread_clear_status_flags ()

```
void
ags_thread_clear_status_flags (AgsThread *thread);
```

Clear status flags.

Parameters

thread	the AgsThread	

Since: 3.0.0

ags_thread_test_sync_tic_flags ()

```
gboolean
ags_thread_test_sync_tic_flags (AgsThread *thread,
                                guint sync_tic_flags);
```

Test *sync_tic_flags* to be set on *thread* .

Parameters

thread	the AgsThread	
sync_tic_flags	the sync-tic flags	

Returns

TRUE if sync-tic flags are set, else FALSE

Since: 3.0.0

ags_thread_set_sync_tic_flags ()

```
void
ags_thread_set_sync_tic_flags (AgsThread *thread,
                               guint sync_tic_flags);
```

Set sync-tic flags.

Parameters

thread	the AgsThread	
sync_tic_flags	the sync-tic flags	

Since: 3.0.0

ags_thread_unset_sync_tic_flags ()

```
void
ags_thread_unset_sync_tic_flags (AgsThread *thread,
                                 guint sync_tic_flags);
```

Unset sync-tic flags.

Parameters

thread	the AgsThread	
sync_tic_flags	the sync-tic flags	

Since: 3.0.0

ags_thread_clear_sync_tic_flags ()

```
void
ags_thread_clear_sync_tic_flags (AgsThread *thread);
```

Clear sync-tic flags.

Parameters

thread	the AgsThread	

Since: 3.0.0

ags_thread_set_current_sync_tic ()

```
void
ags_thread_set_current_sync_tic (AgsThread *thread,
                                 guint current_sync_tic);
```

Set current sync-tic.

Parameters

thread	the AgsThread	
current_sync_tic	the current sync-tic	

Since: 3.0.0

ags_thread_get_current_sync_tic ()

```
guint
ags_thread_get_current_sync_tic (AgsThread *thread);
```

Get current sync-tic.

Parameters

thread	the AgsThread	

Returns

the current sync-tic

Since: 3.0.0

ags_thread_set_delay ()

```
void
ags_thread_set_delay (AgsThread *thread,
                      gdouble delay);
```

Set delay.

Parameters

thread	the AgsThread	
delay	the delay	

Since: 3.0.0

ags_thread_get_delay ()

```
gdouble
ags_thread_get_delay (AgsThread *thread);
```

Get delay.

Parameters

thread	the AgsThread	

Returns

the delay

Since: 3.0.0

ags_thread_set_frequency ()

```
void
ags_thread_set_frequency (AgsThread *thread,
                          gdouble frequency);
```

Set frequency.

Parameters

thread	the AgsThread	
frequency	the frequency	

Since: 3.0.0

ags_thread_get_frequency ()

```
gdouble
ags_thread_get_frequency (AgsThread *thread);
```

Get frequency.

Parameters

thread	the AgsThread	

Returns

the frequency

Since: 3.0.0

ags_thread_set_max_precision ()

```
void
ags_thread_set_max_precision (AgsThread *thread,
                              gdouble max_precision);
```

Set max-precision.

Parameters

thread	the AgsThread	
max_precision	the max-precision	

Since: 3.0.0

ags_thread_get_max_precision ()

```
gdouble
ags_thread_get_max_precision (AgsThread *thread);
```

Get max-precision.

Parameters

thread	the AgsThread	

Returns

the max-precision

Since: 3.0.0

ags_thread_find_type ()

```
AgsThread~*
ags_thread_find_type (AgsThread *thread,
                      GType gtype);
```

Find *gtype* as descendant of *thread* . If its a descendant thread, the ref-count is increased.

Parameters

thread	the AgsThread	
gtype	the GType	

Returns

the matching AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_self ()

```
AgsThread~*
ags_thread_self (void);
```

Thread self.

Returns

the running AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_parent ()

```
AgsThread~*
ags_thread_parent (AgsThread *thread);
```

Get parent thread.

Parameters

thread	the AgsThread

Returns

the parent AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_next ()

```
AgsThread~*
ags_thread_next (AgsThread *thread);
```

Get next thread.

Parameters

thread	the AgsThread

Returns

the next AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_prev ()

```
AgsThread~*
ags_thread_prev (AgsThread *thread);
```

Get prev thread.

Parameters

thread	the AgsThread

Returns

the prev AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_children ()

```
AgsThread~*
ags_thread_children (AgsThread *thread);
```

Get children thread.

Parameters

thread	the AgsThread

Returns

the children AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_get_toplevel ()

```
AgsThread~*
ags_thread_get_toplevel (AgsThread *thread);
```

Retrieve toplevel thread.

Parameters

thread	the AgsThread

Returns

the toplevevel AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_first ()

```
AgsThread~*
ags_thread_first (AgsThread *thread);
```

Retrieve first sibling.

Parameters

thread	the AgsThread

Returns

the very first AgsThread within same tree level.

[transfer full]

Since: 3.0.0

ags_thread_last ()

```
AgsThread~*
ags_thread_last (AgsThread *thread);
```

Retrieve last sibling.

Parameters

thread	the AgsThread

Returns

the very last *AgsThread* within same tree level.

[transfer full]

Since: 3.0.0

ags_thread_lock ()

```
void
ags_thread_lock (AgsThread *thread);
```

Locks the threads own mutex and sets the appropriate flag.

Parameters

thread	the AgsThread

Since: 3.0.0

ags_thread_trylock ()

```
gboolean
ags_thread_trylock (AgsThread *thread);
```

Locks the threads own mutex if available and sets the appropriate flag and returning TRUE. Otherwise return FALSE without lock.

Parameters

thread	the AgsThread	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_thread_unlock ()

```
void
ags_thread_unlock (AgsThread *thread);
```

Unlocks the threads own mutex and unsets the appropriate flag.

Parameters

thread	the AgsThread	

Since: 3.0.0

ags_thread_remove_child ()

```
void
ags_thread_remove_child (AgsThread *thread,
                         AgsThread *child);
```

Remove child of thread.

Parameters

thread	the AgsThread	
child	the child to remove	

Since: 3.0.0

ags_thread_add_child ()

```
void
ags_thread_add_child (AgsThread *thread,
                      AgsThread *child);
```

Add child to thread.

Parameters

thread	the AgsThread	
child	the child to remove	

Since: 3.0.0

ags_thread_add_child_extended ()

```
void
ags_thread_add_child_extended (AgsThread *thread,
                               AgsThread *child,
                               gboolean no_start,
                               gboolean no_wait);
```

Add child to thread.

Parameters

thread	the AgsThread	
child	the child to remove	
no_start	don't start thread	
no_wait	don't wait until started	

Since: 3.0.0

ags_thread_is_current_ready ()

```
gboolean
ags_thread_is_current_ready (AgsThread *current,
                             guint current_sync_tic);
```

ags_thread_is_tree_ready_recursive ()

```
gboolean
ags_thread_is_tree_ready_recursive (AgsThread *thread,
                                    guint current_sync_tic);
```

ags_thread_prepare_current_sync ()

```
void
ags_thread_prepare_current_sync (AgsThread *current,
                                 guint current_sync_tic);
```

ags_thread_prepare_tree_sync_recursive ()

```
void
ags_thread_prepare_tree_sync_recursive
                                (AgsThread *thread,
                                 guint current_sync_tic);
```

ags_thread_set_current_sync ()

```
void
ags_thread_set_current_sync (AgsThread *current,
                             guint current_sync_tic);
```

ags_thread_set_tree_sync_recursive ()

```
void
ags_thread_set_tree_sync_recursive (AgsThread *thread,
                                    guint current_sync_tic);
```

ags_thread_clock ()

```
guint
ags_thread_clock (AgsThread *thread);
```

Clock the thread.

Parameters

thread	the AgsThread instance

Returns

the cycles to be performed

Since: 3.0.0

ags_thread_add_start_queue ()

```
void
ags_thread_add_start_queue (AgsThread *thread,
                            AgsThread *child);
```

Add *child* to *thread* 's start queue.

Parameters

thread	the AgsThread
child	the child AgsThread to start

Since: 3.0.0

ags_thread_add_start_queue_all ()

```
void
ags_thread_add_start_queue_all (AgsThread *thread,
                                GList *child);
```

Add *child* to *thread* 's start queue.

Parameters

thread	the AgsThread	
child	the children as GList containing AgsThread to start.	*[element-type Ags.Thread]*

Since: 3.0.0

ags_thread_start ()

```
void
ags_thread_start (AgsThread *thread);
```

Start the thread.

Parameters

thread	the AgsThread instance	

Since: 3.0.0

ags_thread_run ()

```
void
ags_thread_run (AgsThread *thread);
```

Only for internal use of ags_thread_loop but you may want to set the your very own class function namely your thread's routine.

Parameters

thread	the AgsThread	

Since: 3.0.0

ags_thread_stop ()

```
void
ags_thread_stop (AgsThread *thread);
```

Stop the threads loop by unsetting AGS_THREAD_STATUS_RUNNING flag.

Parameters

thread	the AgsThread	

Since: 3.0.0

ags_thread_new ()

```
AgsThread~*
ags_thread_new ();
```

Create a new instance of AgsThread.

Returns

the new AgsThread

Since: 3.0.0

AGS_IS_THREAD()

```
#define AGS_IS_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_THREAD) ↩
    )
```

AGS_IS_THREAD_CLASS()

```
#define AGS_IS_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_THREAD))
```

AGS_THREAD()

```
#define AGS_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_THREAD,  ↩
    AgsThread))
```

AGS_THREAD_CLASS()

```
#define AGS_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class, AGS_TYPE_THREAD,  ↩
    AgsThreadClass))
```

AGS_THREAD_GET_CLASS()

```
#define AGS_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj, AGS_TYPE_THREAD,  ↩
    AgsThreadClass))
```

ags_thread_flags_get_type ()

```
GType
ags_thread_flags_get_type ();
```

ags_thread_get_type ()

```
GType
ags_thread_get_type ();
```

ags_thread_status_flags_get_type ()

```
GType
ags_thread_status_flags_get_type ();
```

ags_thread_sync_tic_flags_get_type ()

```
GType
ags_thread_sync_tic_flags_get_type ();
```

Types and Values

AGS_THREAD_HERTZ_JIFFIE

```
#define AGS_THREAD_HERTZ_JIFFIE (1000.0)
```

AGS_THREAD_YIELD_JIFFIE

```
#define AGS_THREAD_YIELD_JIFFIE (2.0)
```

AGS_THREAD_DEFAULT_JIFFIE

```
#define AGS_THREAD_DEFAULT_JIFFIE (250.0)
```

AGS_THREAD_DEFAULT_MAX_PRECISION

```
#define AGS_THREAD_DEFAULT_MAX_PRECISION (1000.0)
```

AGS_THREAD_MAX_PRECISION

```
#define AGS_THREAD_MAX_PRECISION (1000.0)
```

AGS_THREAD_DEFAULT_ATTACK

```
#define AGS_THREAD_DEFAULT_ATTACK (1.0)
```

AGS_THREAD_TOLERANCE

```
#define AGS_THREAD_TOLERANCE (0.0)
```

enum AgsThreadFlags

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

Members

AGS_THREAD_UNREF_ON_EXIT	call g_object_unref() before g_thread_exit()
AGS_THREAD_IMMEDIATE_SYNC	do sync immediately
AGS_THREAD_INTERMEDIATE_PRE_SYNC	intermediate pre sync to parent thread
AGS_THREAD_INTERMEDIATE_POST_SYNC	intermediate post sync to parent thread
AGS_THREAD_START_SYNCED_FREQ	sync frequency as starting thread
AGS_THREAD_MARK_SYNCED	mark thread synced
AGS_THREAD_TIME_ACCOUNTING	time accounting causes to track time

enum AgsThreadStatusFlags

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

Members

AGS_THREAD_STATUS_RT_SETUP	realtime setup was performed
AGS_THREAD_STATUS_INITIAL_SYNC	initial sync indicates the thread wasn't synced before
AGS_THREAD_STATUS_INITIAL_RUN	the first call to AgsThread:run()
AGS_THREAD_STATUS_IS_CHAOS_TREE	the thread is not synced
AGS_THREAD_STATUS_START_WAIT	the thread start is waiting
AGS_THREAD_STATUS_START_DONE	the thread start is done
AGS_THREAD_STATUS_READY	the thread is ready
AGS_THREAD_STATUS_WAITING	the thread is waiting
AGS_THREAD_STATUS_RUNNING	the thread is running
AGS_THREAD_STATUS_LOCKED	the thread is locked

AGS_THREAD_STATUS_BUSY	the thread is busy
AGS_THREAD_STATUS_SYNCED	the thread joined the tic based system, it is synced
AGS_THREAD_STATUS_SYNCED_FREQ	the frequency was synced

enum AgsThreadSyncTicFlags

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

Members

AGS_THREAD_SYNC_TIC_WAIT_0	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_0	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_1	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_1	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_2	wait tree to be synced

AGS_THREAD_SYNC_TIC_DONE_2	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_3	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_3	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_4	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_4	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_5	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_5	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_6	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_6	done tree to be synced
AGS_THREAD_SYNC_TIC_WAIT_7	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_7	done tree to be synced

AGS_THREAD_SYNC_TIC_WAIT_8	wait tree to be synced
AGS_THREAD_SYNC_TIC_DONE_8	done tree to be synced

AGS_TYPE_THREAD

```
#define AGS_TYPE_THREAD                 (ags_thread_get_type())
```

AGS_TYPE_THREAD_FLAGS

```
#define AGS_TYPE_THREAD_FLAGS           (ags_thread_flags_get_type())
```

AGS_TYPE_THREAD_STATUS_FLAGS

```
#define AGS_TYPE_THREAD_STATUS_FLAGS    (ags_thread_status_flags_get_type())
```

AGS_TYPE_THREAD_SYNC_TIC_FLAGS

```
#define AGS_TYPE_THREAD_SYNC_TIC_FLAGS (ags_thread_sync_tic_flags_get_type())
```

struct AgsThread

```
struct AgsThread;
```

struct AgsThreadClass

```
struct AgsThreadClass {
  GObjectClass gobject;

  guint (*clock)(AgsThread *thread);

  void (*start)(AgsThread *thread);
  void (*run)(AgsThread *thread);
  void (*stop)(AgsThread *thread);
};
```

Property Details

The "delay" property

```
  "delay"                    double
```

The delay until next tic.

Owner: AgsThread

Flags: Read / Write

Allowed values: [1,1000]

Default value: 1

Since: 3.0.0

The "frequency" property

```
  "frequency"                double
```

The frequency to run at in Hz.

Owner: AgsThread

Flags: Read / Write

Allowed values: [0.01,1000]

Default value: 250

Since: 3.0.0

The "max-precision" property

```
  "max-precision"            double
```

The max-frequency to run at in Hz.

Owner: AgsThread

Flags: Read / Write

Allowed values: [0.01,1000]

Default value: 1000

Since: 3.0.0

Signal Details

The "clock" signal

```
guint
user_function (AgsThread *thread,
               gpointer   user_data)
```

The ::clock() signal is invoked every thread tic.

Parameters

thread	the AgsThread	
user_data	user data set when the signal handler was connected.	

Returns

the number of cycles to perform

Flags: Run Last

Since: 3.0.0

The "run" signal

```
void
user_function (AgsThread *thread,
               gpointer   user_data)
```

The ::run() signal is invoked during run loop.

Parameters

thread	the AgsThread	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "start" signal

```
void
user_function (AgsThread *thread,
               gpointer   user_data)
```

The ::start() signal is invoked as thread started.

Parameters

thread	the AgsThread	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop" signal

```
void
user_function (AgsThread *thread,
               gpointer   user_data)
```

The ::stop() signal is invoked as *thread* stopped.

Parameters

thread	the AgsThread	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 56

AgsThreadApplicationContext

AgsThreadApplicationContext — threaded application context

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_thread_application_context_register_types ()
AgsThreadApplicationContext *	ags_thread_application_context_new ()
#define	AGS_IS_THREAD_APPLICATION_CONTEXT()
#define	AGS_IS_THREAD_APPLICATION_CONTEXT_CLASS()
#define	AGS_THREAD_APPLICATION_CONTEXT()
#define	AGS_THREAD_APPLICATION_CONTEXT_CLASS()
#define	AGS_THREAD_APPLICATION_CONTEXT_GET_CLASS()
GType	ags_thread_application_context_flags_get_type ()
GType	ags_thread_application_context_get_type ()

Properties

AgsThreadPool *	thread-pool	Read / Write

Types and Values

#define	AGS_THREAD_DEFAULT_VERSION
#define	AGS_THREAD_BUILD_ID
enum	AgsThreadApplicationContextFlags
#define	AGS_TYPE_THREAD_APPLICATION_CONTEXT
#define	AGS_TYPE_THREAD_APPLICATION_CONTEXT_FLAGS
struct	AgsThreadApplicationContext
struct	AgsThreadApplicationContextClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsThreadApplicationContextFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsApplicationContext
        &#x2570;&#x2500;&#x2500; AgsThreadApplicationContext
```

Implemented Interfaces

AgsThreadApplicationContext implements AgsConnectable and AgsConcurrencyProvider.

Includes

```
#include <ags/thread/ags_thread_application_context.h>
```

Description

The AgsThreadApplicationContext provides you multi-threaded functionality.

Functions

ags_thread_application_context_register_types ()

```
void
ags_thread_application_context_register_types
                                (AgsApplicationContext *application_context);
```

ags_thread_application_context_new ()

```
AgsThreadApplicationContext~*
ags_thread_application_context_new ();
```

Create a new instance of AgsThreadApplicationContext.

Returns

the new AgsThreadApplicationContext

Since: 3.0.0

AGS_IS_THREAD_APPLICATION_CONTEXT()

```
#define AGS_IS_THREAD_APPLICATION_CONTEXT(obj)             (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_THREAD_APPLICATION_CONTEXT))
```

AGS_IS_THREAD_APPLICATION_CONTEXT_CLASS()

```
#define AGS_IS_THREAD_APPLICATION_CONTEXT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_THREAD_APPLICATION_CONTEXT))
```

AGS_THREAD_APPLICATION_CONTEXT()

```
#define AGS_THREAD_APPLICATION_CONTEXT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_THREAD_APPLICATION_CONTEXT, AgsThreadApplicationContext))
```

AGS_THREAD_APPLICATION_CONTEXT_CLASS()

```
#define AGS_THREAD_APPLICATION_CONTEXT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_THREAD_APPLICATION_CONTEXT, AgsThreadApplicationContextClass))
```

AGS_THREAD_APPLICATION_CONTEXT_GET_CLASS()

```
#define AGS_THREAD_APPLICATION_CONTEXT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_THREAD_APPLICATION_CONTEXT, AgsThreadApplicationContextClass))
```

ags_thread_application_context_flags_get_type ()

```
GType
ags_thread_application_context_flags_get_type
                                ();
```

ags_thread_application_context_get_type ()

```
GType
ags_thread_application_context_get_type
                                ();
```

Types and Values

AGS_THREAD_DEFAULT_VERSION

```
#define AGS_THREAD_DEFAULT_VERSION "0.7.122.8"
```

AGS_THREAD_BUILD_ID

```
#define AGS_THREAD_BUILD_ID "Tue Mar 21 20:03:29 CET 2017"
```

enum AgsThreadApplicationContextFlags

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

Members

AGS_THREAD_APPLICATION_CONTEXT_SINGLE_THREAD	run all threads in one single loop

AGS_TYPE_THREAD_APPLICATION_CONTEXT

```
#define AGS_TYPE_THREAD_APPLICATION_CONTEXT                ( ↩
    ags_thread_application_context_get_type())
```

AGS_TYPE_THREAD_APPLICATION_CONTEXT_FLAGS

```
#define AGS_TYPE_THREAD_APPLICATION_CONTEXT_FLAGS          ( ↩
    ags_thread_application_context_flags_get_type())
```

struct AgsThreadApplicationContext

```
struct AgsThreadApplicationContext;
```

struct AgsThreadApplicationContextClass

```
struct AgsThreadApplicationContextClass {
  AgsApplicationContextClass application_context;
};
```

Property Details

The "thread-pool" property

```
  "thread-pool"              AgsThreadPool~*
```

The assigned thread pool.

Owner: AgsThreadApplicationContext

Flags: Read / Write

Since: 3.0.0

Chapter 57

AgsThreadPool

AgsThreadPool — thread pool

Stability Level

Stable, unless otherwise indicated

Functions

AgsThread *	ags_thread_pool_pull ()
void	ags_thread_pool_start ()
AgsThreadPool *	ags_thread_pool_new ()
#define	AGS_IS_THREAD_POOL()
#define	AGS_IS_THREAD_POOL_CLASS()
#define	AGS_THREAD_POOL()
#define	AGS_THREAD_POOL_CLASS()
#define	AGS_THREAD_POOL_GET_CLASS()
GType	ags_thread_pool_flags_get_type ()
GType	ags_thread_pool_get_type ()

Properties

guint	max-threads	Read / Write
guint	max-unused-threads	Read / Write

Signals

void	start	Run Last

Types and Values

#define	AGS_THREAD_POOL_DEFAULT_MAX_UNUSED_THREADS
#define	AGS_THREAD_POOL_DEFAULT_MAX_THREADS
enum	AgsThreadPoolFlags
#define	AGS_TYPE_THREAD_POOL
#define	AGS_TYPE_THREAD_POOL_FLAGS
struct	AgsThreadPool
struct	AgsThreadPoolClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsThreadPoolFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsThreadPool
```

Includes

```
#include <ags/thread/ags_thread_pool.h>
```

Description

The AgsThreadPool acts as preinstantiated threads store. This can achieve enormeous performance.

Functions

ags_thread_pool_pull ()

```
AgsThread~*
ags_thread_pool_pull (AgsThreadPool *thread_pool);
```

Pull a previously instantiated AgsReturnableThread. Note this function may block until a new thread is available.

Parameters

thread_pool	the AgsThreadPool

Returns

a new AgsThread.

[transfer full]

Since: 3.0.0

ags_thread_pool_start ()

```
void
ags_thread_pool_start (AgsThreadPool *thread_pool);
```

Start the thread pool.

Parameters

thread_pool	the AgsThreadPool

Since: 3.0.0

ags_thread_pool_new ()

```
AgsThreadPool~*
ags_thread_pool_new (AgsThread *parent);
```

Create a new AgsThreadPool.

Parameters

parent	the parent AgsThread of returnable threads

Returns

the new AgsThreadPool

Since: 3.0.0

AGS_IS_THREAD_POOL()

```
#define AGS_IS_THREAD_POOL(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_THREAD_POOL))
```

AGS_IS_THREAD_POOL_CLASS()

```
#define AGS_IS_THREAD_POOL_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_THREAD_POOL))
```

AGS_THREAD_POOL()

```
#define AGS_THREAD_POOL(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_THREAD_POOL, AgsThreadPool))
```

AGS_THREAD_POOL_CLASS()

```
#define AGS_THREAD_POOL_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_THREAD_POOL, AgsThreadPoolClass))
```

AGS_THREAD_POOL_GET_CLASS()

```
#define AGS_THREAD_POOL_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_THREAD_POOL, AgsThreadPoolClass))
```

ags_thread_pool_flags_get_type ()

```
GType
ags_thread_pool_flags_get_type ();
```

ags_thread_pool_get_type ()

```
GType
ags_thread_pool_get_type ();
```

Types and Values

AGS_THREAD_POOL_DEFAULT_MAX_UNUSED_THREADS

```
#define AGS_THREAD_POOL_DEFAULT_MAX_UNUSED_THREADS (8)
```

AGS_THREAD_POOL_DEFAULT_MAX_THREADS

```
#define AGS_THREAD_POOL_DEFAULT_MAX_THREADS (1024)
```

enum AgsThreadPoolFlags

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

Members

AGS_THREAD_POOL_RUNNING	the thread pool is running

AGS_THREAD_POOL_RT_SETUP	realtime setup has been performed

AGS_TYPE_THREAD_POOL

```
#define AGS_TYPE_THREAD_POOL                (ags_thread_pool_get_type())
```

AGS_TYPE_THREAD_POOL_FLAGS

```
#define AGS_TYPE_THREAD_POOL_FLAGS          (ags_thread_pool_flags_get_type())
```

struct AgsThreadPool

```
struct AgsThreadPool;
```

struct AgsThreadPoolClass

```
struct AgsThreadPoolClass {
  GObjectClass gobject;

  void (*start)(AgsThreadPool *thread_pool);
};
```

Property Details

The "max-threads" property

```
  "max-threads"              guint
```

The maximum amount of threads available.

Owner: AgsThreadPool

Flags: Read / Write

Allowed values: [1,65535]

Default value: 1024

Since: 3.0.0

The "max-unused-threads" property

```
"max-unused-threads"       guint
```

The maximum amount of unused threads available.

Owner: AgsThreadPool

Flags: Read / Write

Allowed values: [1,65535]

Default value: 8

Since: 3.0.0

Signal Details

The "start" signal

```
void
user_function (AgsThreadPool *thread_pool,
               gpointer       user_data)
```

The ::start() signal is invoked in order to started the pool.

Parameters

thread_pool	the AgsThreadPool	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 58

AgsTimestamp

AgsTimestamp — Timestamp unix and alike

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_TIMESTAMP_GET_OBJ_MUTEX()
gboolean	ags_timestamp_test_flags ()
void	ags_timestamp_set_flags ()
void	ags_timestamp_unset_flags ()
time_t	ags_timestamp_get_unix_time ()
void	ags_timestamp_set_unix_time ()
guint64	ags_timestamp_get_ags_offset ()
void	ags_timestamp_set_ags_offset ()
AgsTimestamp *	ags_timestamp_new ()
#define	AGS_IS_TIMESTAMP()
#define	AGS_IS_TIMESTAMP_CLASS()
#define	AGS_TIMESTAMP()
#define	AGS_TIMESTAMP_CLASS()
#define	AGS_TIMESTAMP_GET_CLASS()
GType	ags_timestamp_flags_get_type ()
GType	ags_timestamp_get_type ()

Types and Values

enum	AgsTimestampFlags
#define	AGS_TYPE_TIMESTAMP
#define	AGS_TYPE_TIMESTAMP_FLAGS
struct	AgsTimestamp
struct	AgsTimestampClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsTimestampFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsTimestamp
```

Includes

```
#include <ags/thread/ags_timestamp.h>
```

Description

AgsTimestamp measure of time.

Functions

AGS_TIMESTAMP_GET_OBJ_MUTEX()

```
#define AGS_TIMESTAMP_GET_OBJ_MUTEX(obj) (&(((AgsTimestamp *) obj)->obj_mutex))
```

ags_timestamp_test_flags ()

```
gboolean
ags_timestamp_test_flags (AgsTimestamp *timestamp,
                          guint flags);
```

Test *flags* to be set.

Parameters

timestamp	the AgsTimestamp	
flags	the flags	

Returns

if *flags* set returning TRUE otherwise FALSE

Since: 3.0.0

ags_timestamp_set_flags ()

```
void
ags_timestamp_set_flags (AgsTimestamp *timestamp,
                         guint flags);
```

Set *flags* of *timestamp* .

Parameters

timestamp	the AgsTimestamp	
flags	the flags	

Since: 3.0.0

ags_timestamp_unset_flags ()

```
void
ags_timestamp_unset_flags (AgsTimestamp *timestamp,
                           guint flags);
```

Unset *flags* of *timestamp* .

Parameters

timestamp	the AgsTimestamp	
flags	the flags	

Since: 3.0.0

ags_timestamp_get_unix_time ()

```
time_t
ags_timestamp_get_unix_time (AgsTimestamp *timestamp);
```

Get unix time.

Parameters

timestamp	the AgsTimestamp	

Returns

the unix time as time_t value

Since: 3.0.0

ags_timestamp_set_unix_time ()

```
void
ags_timestamp_set_unix_time (AgsTimestamp *timestamp,
                             time_t unix_time);
```

Set unix time.

Parameters

timestamp	the AgsTimestamp	
unix_time	the unix time value	

Since: 3.0.0

ags_timestamp_get_ags_offset ()

```
guint64
ags_timestamp_get_ags_offset (AgsTimestamp *timestamp);
```

Get AGS offset.

Parameters

timestamp	the AgsTimestamp	

Returns

the AGS offset as unsigned 64 bit integer

Since: 3.0.0

ags_timestamp_set_ags_offset ()

```
void
ags_timestamp_set_ags_offset (AgsTimestamp *timestamp,
                              guint64 ags_offset);
```

Set AGS offset as unsigned 64 bit integer.

Parameters

timestamp	the AgsTimestamp	
ags_offset	the AGS offset	

Since: 3.0.0

ags_timestamp_new ()

```
AgsTimestamp~*
ags_timestamp_new ();
```

Creates an AgsTimestamp

Returns

a new AgsTimestamp

Since: 3.0.0

AGS_IS_TIMESTAMP()

```
#define AGS_IS_TIMESTAMP(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_TIMESTAMP))
```

AGS_IS_TIMESTAMP_CLASS()

```
#define AGS_IS_TIMESTAMP_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_TIMESTAMP))
```

AGS_TIMESTAMP()

```
#define AGS_TIMESTAMP(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_TIMESTAMP, AgsTimestamp))
```

AGS_TIMESTAMP_CLASS()

```
#define AGS_TIMESTAMP_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_TIMESTAMP, AgsTimestampClass))
```

AGS_TIMESTAMP_GET_CLASS()

```
#define AGS_TIMESTAMP_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_TIMESTAMP, AgsTimestampClass))
```

ags_timestamp_flags_get_type ()

```
GType
ags_timestamp_flags_get_type (void);
```

ags_timestamp_get_type ()

```
GType
ags_timestamp_get_type (void);
```

Types and Values

enum AgsTimestampFlags

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

Members

AGS_TIMESTAMP_UNIX	unix times-tamp is used
AGS_TIMESTAMP_OFFSET	internal off-set is used
AGS_TIMESTAMP_OUTDATED	the times-tamp is out-dated

AGS_TYPE_TIMESTAMP

```
#define AGS_TYPE_TIMESTAMP                (ags_timestamp_get_type())
```

AGS_TYPE_TIMESTAMP_FLAGS

```
#define AGS_TYPE_TIMESTAMP_FLAGS          (ags_timestamp_flags_get_type())
```

struct AgsTimestamp

```
struct AgsTimestamp;
```

struct AgsTimestampClass

```
struct AgsTimestampClass {
  GObjectClass gobject;
};
```

Chapter 59

AgsWorkerThread

AgsWorkerThread — Worker thread class discarding base frequency

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_worker_thread_test_status_flags ()
void	ags_worker_thread_set_status_flags ()
void	ags_worker_thread_unset_status_flags ()
void *	ags_woker_thread_do_poll_loop ()
void	ags_worker_thread_do_poll ()
AgsWorkerThread *	ags_worker_thread_new ()
#define	AGS_IS_WORKER_THREAD()
#define	AGS_IS_WORKER_THREAD_CLASS()
#define	AGS_WORKER_THREAD()
#define	AGS_WORKER_THREAD_CLASS()
#define	AGS_WORKER_THREAD_GET_CLASS()
GType	ags_worker_thread_get_type ()
GType	ags_worker_thread_status_flags_get_type ()

Signals

void	do-poll	Run Last

Types and Values

#define	AGS_WORKER_THREAD_DEFAULT_JIFFIE
enum	AgsWorkerThreadStatusFlags
#define	AGS_TYPE_WORKER_THREAD
#define	AGS_TYPE_WORKER_THREAD_STATUS_FLAGS

struct	AgsWorkerThread
struct	AgsWorkerThreadClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsWorkerThreadStatusFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsThread
        &#x2570;&#x2500;&#x2500; AgsWorkerThread
            &#x2570;&#x2500;&#x2500; AgsDestroyWorker
```

Implemented Interfaces

AgsWorkerThread implements AgsConnectable.

Includes

```
#include <ags/thread/ags_worker_thread.h>
```

Description

The AgsWorkerThread does non-realtime work. You might want to synchronize to the run signal within your ::do_poll() method.

You usually connect to AgsThread::do_poll() event. The poll event is invoked as long as AGS_WORKER_THREAD_STATUS_RUNNIN status flag is set.

You might want to inject to a thread tree using a AgsTask implementation.

Functions

ags_worker_thread_test_status_flags ()

```
gboolean
ags_worker_thread_test_status_flags (AgsWorkerThread *worker_thread,
                                     guint status_flags);
```

Test *status_flags* of *worker_thread* .

Parameters

worker_thread	the AgsWorkerThread	
status_flags	status flags	

Returns

TRUE if status flags set, otherwise FALSE

Since: 3.0.0

ags_worker_thread_set_status_flags ()

```
void
ags_worker_thread_set_status_flags (AgsWorkerThread *worker_thread,
                                    guint status_flags);
```

Set status flags.

Parameters

worker_thread	the AgsWorkerThread	
status_flags	status flags	

Since: 3.0.0

ags_worker_thread_unset_status_flags ()

```
void
ags_worker_thread_unset_status_flags (AgsWorkerThread *worker_thread,
                                      guint status_flags);
```

Unset status flags.

Parameters

worker_thread	the AgsWorkerThread	
status_flags	status flags	

Since: 3.0.0

ags_woker_thread_do_poll_loop ()

```
void~*
ags_woker_thread_do_poll_loop (void *ptr);
```

Do loop and invoke ags_worker_thread_do_poll() unless flag AGS_WORKER_THREAD_RUNNING was unset.

Parameters

ptr	the AgsWorkerThread	

Since: 3.0.0

ags_worker_thread_do_poll ()

```
void
ags_worker_thread_do_poll (AgsWorkerThread *worker_thread);
```

Do poll your work. It is called of the worker thread.

Parameters

worker_thread	the AgsWorkerThread

Since: 3.0.0

ags_worker_thread_new ()

```
AgsWorkerThread~*
ags_worker_thread_new ();
```

Create a new instance of AgsWorkerThread.

Returns

the new AgsWorkerThread

Since: 3.0.0

AGS_IS_WORKER_THREAD()

```
#define AGS_IS_WORKER_THREAD(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_WORKER_THREAD))
```

AGS_IS_WORKER_THREAD_CLASS()

```
#define AGS_IS_WORKER_THREAD_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_WORKER_THREAD))
```

AGS_WORKER_THREAD()

```
#define AGS_WORKER_THREAD(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_WORKER_THREAD, AgsWorkerThread))
```

AGS_WORKER_THREAD_CLASS()

```
#define AGS_WORKER_THREAD_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_WORKER_THREAD, AgsWorkerThreadClass))
```

AGS_WORKER_THREAD_GET_CLASS()

```
#define AGS_WORKER_THREAD_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_WORKER_THREAD, AgsWorkerThreadClass))
```

ags_worker_thread_get_type ()

```
GType
ags_worker_thread_get_type ();
```

ags_worker_thread_status_flags_get_type ()

```
GType
ags_worker_thread_status_flags_get_type
                                ();
```

Types and Values

AGS_WORKER_THREAD_DEFAULT_JIFFIE

```
#define AGS_WORKER_THREAD_DEFAULT_JIFFIE (AGS_THREAD_DEFAULT_MAX_PRECISION)
```

enum AgsWorkerThreadStatusFlags

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

Members

AGS_WORKER_THREAD_STATUS_RUNNING	the worker is running
AGS_WORKER_THREAD_STATUS_RUN_WAIT	sync wait
AGS_WORKER_THREAD_STATUS_RUN_DONE	sync done
AGS_WORKER_THREAD_STATUS_RUN_SYNC	do sync

AGS_TYPE_WORKER_THREAD

```
#define AGS_TYPE_WORKER_THREAD                (ags_worker_thread_get_type())
```

AGS_TYPE_WORKER_THREAD_STATUS_FLAGS

```
#define AGS_TYPE_WORKER_THREAD_STATUS_FLAGS   (ags_worker_thread_status_flags_get_type())
```

struct AgsWorkerThread

```
struct AgsWorkerThread;
```

struct AgsWorkerThreadClass

```
struct AgsWorkerThreadClass {
  AgsThreadClass thread;

  void (*do_poll)(AgsWorkerThread *worker_thread);
};
```

Signal Details

The "do-poll" signal

```
void
user_function (AgsWorkerThread *thread,
               gpointer         user_data)
```

The ::do_poll() signal runs independently of ::run() but might be synchronized using a conditional lock.

Parameters

thread	the AgsWorkerThread	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Part VI

XML-RPC - server

Chapter 60

AgsRegistry

AgsRegistry — remote control registry

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_REGISTRY_GET_OBJ_MUTEX()
AgsRegistryEntry *	ags_registry_entry_alloc ()
void	ags_registry_entry_free ()
void	ags_registry_add_entry ()
AgsRegistryEntry *	ags_registry_find_entry ()
AgsRegistry *	ags_registry_new ()
#define	AGS_IS_REGISTRY()
#define	AGS_IS_REGISTRY_CLASS()
#define	AGS_REGISTRY()
#define	AGS_REGISTRY_CLASS()
#define	AGS_REGISTRY_GET_CLASS()
GType	ags_registry_get_type ()

Properties

AgsServer *	server	Read / Write

Types and Values

#define	AGS_TYPE_REGISTRY_FLAGS
struct	AgsRegistryEntry
#define	AGS_TYPE_REGISTRY
struct	AgsRegistry
struct	AgsRegistryClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsRegistry
```

Includes

```
#include <ags/server/ags_registry.h>
```

Description

The AgsRegistry is a registry where you are able to lookup objects.

Functions

AGS_REGISTRY_GET_OBJ_MUTEX()

```
#define AGS_REGISTRY_GET_OBJ_MUTEX(obj) (&(((AgsRegistry *) obj)->obj_mutex))
```

ags_registry_entry_alloc ()

```
AgsRegistryEntry~*
ags_registry_entry_alloc ();
```

Allocated AgsRegistryEntry.

Returns

the newly allocated AgsRegistryEntry

Since: 3.0.0

ags_registry_entry_free ()

```
void
ags_registry_entry_free (AgsRegistryEntry *registry_entry);
```

Free *registry_entry*

Parameters

registry_entry	the AgsRegistryEntry

Since: 3.0.0

ags_registry_add_entry ()

```
void
ags_registry_add_entry (AgsRegistry *registry,
                        AgsRegistryEntry *registry_entry);
```

Add *registry_entry* to *registry* .

Parameters

registry	the AgsRegistry	
registry_entry	the AgsRegistryEntry to add	

Since: 3.0.0

ags_registry_find_entry ()

```
AgsRegistryEntry~*
ags_registry_find_entry (AgsRegistry *registry,
                         AgsUUID *id);
```

Find *id* as AgsRegistryEntry in *registry* .

Parameters

registry	the AgsRegistry	
id	the AgsUUID to find	

Since: 3.0.0

ags_registry_new ()

```
AgsRegistry~*
ags_registry_new ();
```

AGS_IS_REGISTRY()

```
#define AGS_IS_REGISTRY(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_REGISTRY))
```

AGS_IS_REGISTRY_CLASS()

```
#define AGS_IS_REGISTRY_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_REGISTRY))
```

AGS_REGISTRY()

```
#define AGS_REGISTRY(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_REGISTRY, AgsRegistry))
```

AGS_REGISTRY_CLASS()

```
#define AGS_REGISTRY_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class, AGS_TYPE_REGISTRY, ↩
     AgsRegistryClass))
```

AGS_REGISTRY_GET_CLASS()

```
#define AGS_REGISTRY_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj, AGS_TYPE_REGISTRY, ↩
     AgsRegistryClass))
```

ags_registry_get_type ()

```
GType
ags_registry_get_type ();
```

Types and Values

AGS_TYPE_REGISTRY_FLAGS

```
#define AGS_TYPE_REGISTRY_FLAGS          (ags_registry_flags_get_type())
```

struct AgsRegistryEntry

```
struct AgsRegistryEntry {
  AgsUUID *id;
  GValue *entry;
};
```

AgsRegistryEntry is an entry that you might want to lookup remotely.

Members

AgsUUID *id;	the AgsUUID
GValue *entry;	the actual entry

AGS_TYPE_REGISTRY

```
#define AGS_TYPE_REGISTRY                (ags_registry_get_type())
```

struct AgsRegistry

```
struct AgsRegistry;
```

struct AgsRegistryClass

```
struct AgsRegistryClass {
  GObjectClass gobject;
};
```

Property Details

The "server" property

```
  "server"                   AgsServer~*
```

The assigned AgsServer

Owner: AgsRegistry

Flags: Read / Write

Since: 3.0.0

Chapter 61

AgsServer

AgsServer — Remote control server

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SERVER_GET_OBJ_MUTEX()
gboolean	ags_server_test_flags ()
void	ags_server_set_flags ()
void	ags_server_unset_flags ()
AgsServerInfo *	ags_server_info_alloc ()
void	ags_server_add_controller ()
void	ags_server_remove_controller ()
void	ags_server_start ()
void	ags_server_stop ()
gboolean	ags_server_listen ()
AgsServer *	ags_server_new ()
#define	AGS_IS_SERVER()
#define	AGS_IS_SERVER_CLASS()
#define	AGS_SERVER()
#define	AGS_SERVER_CLASS()
#define	AGS_SERVER_GET_CLASS()
GType	ags_server_flags_get_type ()
GType	ags_server_get_type ()

Properties

gpointer	controller	Read / Write
char *	domain	Read / Write
AgsFrontController *	front-controller	Read / Write
char *	ip4	Read / Write
char *	ip6	Read / Write

char *	path	Read / Write
char *	realm	Read / Write
guint	server-port	Read / Write

Signals

gboolean	listen	Run Last
void	start	Run Last
void	stop	Run Last

Types and Values

#define	AGS_SERVER_DEFAULT_SERVER_PORT
#define	AGS_SERVER_DEFAULT_DOMAIN
#define	AGS_SERVER_DEFAULT_INET4_ADDRESS
#define	AGS_SERVER_DEFAULT_INET6_ADDRESS
#define	AGS_SERVER_DEFAULT_AUTH_MODULE
#define	AGS_SERVER_DEFAULT_BACKLOG
enum	AgsServerFlags
struct	AgsServerInfo
#define	AGS_TYPE_SERVER
#define	AGS_TYPE_SERVER_FLAGS
struct	AgsServer
struct	AgsServerClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsServerFlags
    GObject
    &#x2570;&#x2500;&#x2500; AgsServer
```

Includes

```
#include <ags/server/ags_server.h>
```

Description

The AgsServer is a XML-RPC server supporting authentication. See AgsXmlPasswordStore for a built-in authentication module using XML password files.

Functions

AGS_SERVER_GET_OBJ_MUTEX()

```
#define AGS_SERVER_GET_OBJ_MUTEX(obj) (&(((AgsServer *) obj)->obj_mutex))
```

ags_server_test_flags ()

```
gboolean
ags_server_test_flags (AgsServer *server,
                       guint flags);
```

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

Parameters

server	the AgsServer	
flags	the flags	

Returns

TRUE if flags are set, else FALSE

Since: 3.0.0

ags_server_set_flags ()

```
void
ags_server_set_flags (AgsServer *server,
                      guint flags);
```

Enable a feature of *server* .

Parameters

server	the AgsServer	
flags	see AgsServerFlags	

Since: 3.0.0

ags_server_unset_flags ()

```
void
ags_server_unset_flags (AgsServer *server,
                        guint flags);
```

Disable a feature of *server* .

Parameters

server	the AgsServer	
flags	see AgsServerFlags	

Since: 3.0.0

ags_server_info_alloc ()

```
AgsServerInfo~*
ags_server_info_alloc (gchar *server_name,
                       gchar *uuid);
```

Allocate server info.

Parameters

server_name	the server name	
uuid	the uuid	

Returns

the allocated AgsServerInfo

Since: 3.0.0

ags_server_add_controller ()

```
void
ags_server_add_controller (AgsServer *server,
                           GObject *controller);
```

Add *controller* to *server* .

Parameters

server	the AgsServer	
controller	the AgsController	

Since: 3.0.0

ags_server_remove_controller ()

```
void
ags_server_remove_controller (AgsServer *server,
                              GObject *controller);
```

Remove *controller* from *server* .

Parameters

server	the AgsServer	
controller	the AgsController	

Since: 3.0.0

ags_server_start ()

```
void
ags_server_start (AgsServer *server);
```

Start the XMLRPC-C abyss server.

Parameters

server	the AgsServer

Since: 3.0.0

ags_server_stop ()

```
void
ags_server_stop (AgsServer *server);
```

Stop the XMLRPC-C abyss server.

Parameters

server	the AgsServer

Since: 3.0.0

ags_server_listen ()

```
gboolean
ags_server_listen (AgsServer *server);
```

Listen as server.

Parameters

server	the AgsServer

Returns

TRUE as a new connection was initiated, otherwise FALSE

Since: 3.0.0

ags_server_new ()

```
AgsServer~*
ags_server_new ();
```

Instantiate AgsServer.

Returns

a new AgsServer

Since: 3.0.0

AGS_IS_SERVER()

```
#define AGS_IS_SERVER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj), AGS_TYPE_SERVER) ↩
    )
```

AGS_IS_SERVER_CLASS()

```
#define AGS_IS_SERVER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class), AGS_TYPE_SERVER))
```

AGS_SERVER()

```
#define AGS_SERVER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj), AGS_TYPE_SERVER,  ↩
    AgsServer))
```

AGS_SERVER_CLASS()

```
#define AGS_SERVER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class, AGS_TYPE_SERVER,  ↩
    AgsServerClass))
```

AGS_SERVER_GET_CLASS()

```
#define AGS_SERVER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj, AGS_TYPE_SERVER,  ↩
    AgsServerClass))
```

ags_server_flags_get_type ()

```
GType
ags_server_flags_get_type ();
```

ags_server_get_type ()

```
GType
ags_server_get_type ();
```

Types and Values

AGS_SERVER_DEFAULT_SERVER_PORT

```
#define AGS_SERVER_DEFAULT_SERVER_PORT (8080)
```

AGS_SERVER_DEFAULT_DOMAIN

```
#define AGS_SERVER_DEFAULT_DOMAIN "localhost"
```

AGS_SERVER_DEFAULT_INET4_ADDRESS

```
#define AGS_SERVER_DEFAULT_INET4_ADDRESS "127.0.0.1"
```

AGS_SERVER_DEFAULT_INET6_ADDRESS

```
#define AGS_SERVER_DEFAULT_INET6_ADDRESS "::1"
```

AGS_SERVER_DEFAULT_AUTH_MODULE

```
#define AGS_SERVER_DEFAULT_AUTH_MODULE "ags-xml-authentication"
```

AGS_SERVER_DEFAULT_BACKLOG

```
#define AGS_SERVER_DEFAULT_BACKLOG (512)
```

enum AgsServerFlags

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

Members

AGS_SERVER_STARTED	the server was started
AGS_SERVER_RUNNING	the server is up and running

AGS_SERVER_TERMINATING	the server is closing connections and terminating
AGS_SERVER_INET4	use IPv4
AGS_SERVER_INET6	use IPv6
AGS_SERVER_ANY_ADDRESS	listen on any address
AGS_SERVER_AUTO_START	start the server

struct AgsServerInfo

```
struct AgsServerInfo {
  gchar *uuid;
  gchar *server_name;
};
```

AgsServerInfo does identify the server and provides some basic information about it.

Members

gchar *uuid;	the assigned uuid
gchar *server_name;	the server's name

AGS_TYPE_SERVER

```
#define AGS_TYPE_SERVER                 (ags_server_get_type())
```

AGS_TYPE_SERVER_FLAGS

```
#define AGS_TYPE_SERVER_FLAGS           (ags_server_flags_get_type())
```

struct AgsServer

```
struct AgsServer;
```

struct AgsServerClass

```
struct AgsServerClass {
  GObjectClass gobject;

  void (*start)(AgsServer *server);
  void (*stop)(AgsServer *server);

  gboolean (*listen)(AgsServer *server);
};
```

Property Details

The "controller" property

```
  "controller"                gpointer
```

The assigned AgsController providing default settings.

[transfer full]

Owner: AgsServer

Flags: Read / Write

Since: 3.0.0

The "domain" property

```
  "domain"                    char~*
```

The domain to use.

Owner: AgsServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "front-controller" property

```
  "front-controller"          AgsFrontController~*
```

The assigned AgsFrontController.

Owner: AgsServer

Flags: Read / Write

Since: 3.0.0

The "ip4" property

```
"ip4"                    char~*
```

The IPv4 address as string of the server.

Owner: AgsServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "ip6" property

```
"ip6"                    char~*
```

The IPv6 address as string of the server.

Owner: AgsServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "path" property

```
"path"                   char~*
```

The path to use.

Owner: AgsServer

Flags: Read / Write

Default value: NULL

Since: 4.0.0

The "realm" property

```
"realm"                  char~*
```

The realm to use.

Owner: AgsServer

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "server-port" property

```
  "server-port"              guint
```

The server port to use.

Owner: AgsServer

Flags: Read / Write

Default value: 8080

Since: 3.0.0

Signal Details

The "listen" signal

```
gboolean
user_function (AgsServer *server,
               gpointer   user_data)
```

The ::listen signal is emited during listen of server.

Parameters

server	the AgsServer	
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 (AgsServer *server,
               gpointer   user_data)
```

The ::start signal is emitted as the server starts.

Parameters

server	the AgsServer	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

The "stop" signal

```
void
user_function (AgsServer *server,
               gpointer   user_data)
```

The ::stop signal is emitted as the server stops.

Parameters

server	the AgsServer	
user_data	user data set when the signal handler was connected.	

Flags: Run Last

Since: 3.0.0

Chapter 62

AgsServerApplicationContext

AgsServerApplicationContext —

Stability Level

Stable, unless otherwise indicated

Functions

AgsServerApplicationContext *	ags_server_application_context_new ()
#define	AGS_IS_SERVER_APPLICATION_CONTEXT()
#define	AGS_IS_SERVER_APPLICATION_CONTEXT_CLASS()
#define	AGS_SERVER_APPLICATION_CONTEXT()
#define	AGS_SERVER_APPLICATION_CONTEXT_CLASS()
#define	AGS_SERVER_APPLICATION_CONTEXT_GET_CLASS()
GType	ags_server_application_context_get_type ()

Types and Values

#define	AGS_SERVER_VERSION
#define	AGS_SERVER_BUILD_ID
#define	AGS_TYPE_SERVER_APPLICATION_CONTEXT
struct	AgsServerApplicationContext
struct	AgsServerApplicationContextClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsApplicationContext
    &#x2570;&#x2500;&#x2500; AgsServerApplicationContext
```

Implemented Interfaces

AgsServerApplicationContext implements AgsConnectable, AgsConcurrencyProvider and AgsServiceProvider.

Description

Functions

ags_server_application_context_new ()

```
AgsServerApplicationContext~*
ags_server_application_context_new ();
```

Instantiate server application context.

Returns

the new AgsServerApplicationContext

Since: 3.0.0

AGS_IS_SERVER_APPLICATION_CONTEXT()

```
#define AGS_IS_SERVER_APPLICATION_CONTEXT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE (( ↩
    obj), AGS_TYPE_SERVER_APPLICATION_CONTEXT))
```

AGS_IS_SERVER_APPLICATION_CONTEXT_CLASS()

```
#define AGS_IS_SERVER_APPLICATION_CONTEXT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class ↩
    ), AGS_TYPE_SERVER_APPLICATION_CONTEXT))
```

AGS_SERVER_APPLICATION_CONTEXT()

```
#define AGS_SERVER_APPLICATION_CONTEXT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj ↩
    ), AGS_TYPE_SERVER_APPLICATION_CONTEXT, AgsServerApplicationContext))
```

AGS_SERVER_APPLICATION_CONTEXT_CLASS()

```
#define AGS_SERVER_APPLICATION_CONTEXT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_SERVER_APPLICATION_CONTEXT, AgsServerApplicationContextClass))
```

AGS_SERVER_APPLICATION_CONTEXT_GET_CLASS()

```
#define AGS_SERVER_APPLICATION_CONTEXT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_SERVER_APPLICATION_CONTEXT, AgsServerApplicationContextClass))
```

ags_server_application_context_get_type ()

```
GType
ags_server_application_context_get_type
                                    ();
```

Types and Values

AGS_SERVER_VERSION

```
#define AGS_SERVER_VERSION "3.0.0"
```

AGS_SERVER_BUILD_ID

```
#define AGS_SERVER_BUILD_ID "Sat Dec  7 10:02:36 UTC 2019"
```

AGS_TYPE_SERVER_APPLICATION_CONTEXT

```
#define AGS_TYPE_SERVER_APPLICATION_CONTEXT                ( ↩
    ags_server_application_context_get_type())
```

struct AgsServerApplicationContext

```
struct AgsServerApplicationContext;
```

struct AgsServerApplicationContextClass

```
struct AgsServerApplicationContextClass {
  AgsApplicationContextClass application_context;
};
```

Chapter 63

AgsServerStatus

AgsServerStatus —

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SERVER_STATUS_GET_OBJ_MUTEX()
AgsServerStatus *	ags_server_status_new ()
#define	AGS_IS_SERVER_STATUS()
#define	AGS_IS_SERVER_STATUS_CLASS()
#define	AGS_SERVER_STATUS()
#define	AGS_SERVER_STATUS_CLASS()
#define	AGS_SERVER_STATUS_GET_CLASS()
GType	ags_server_status_get_type ()

Types and Values

#define	AGS_TYPE_SERVER_STATUS
struct	AgsServerStatus
struct	AgsServerStatusClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsServerStatus
```

Implemented Interfaces

AgsServerStatus implements AgsConnectable.

Description

Functions

AGS_SERVER_STATUS_GET_OBJ_MUTEX()

```
#define AGS_SERVER_STATUS_GET_OBJ_MUTEX(obj) (&(((AgsServerStatus *) obj)->obj_mutex))
```

ags_server_status_new ()

```
AgsServerStatus~*
ags_server_status_new ();
```

Instantiate AgsServerStatus.

Returns

a new AgsServerStatus

Since: 3.0.0

AGS_IS_SERVER_STATUS()

```
#define AGS_IS_SERVER_STATUS(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SERVER_STATUS))
```

AGS_IS_SERVER_STATUS_CLASS()

```
#define AGS_IS_SERVER_STATUS_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SERVER_STATUS))
```

AGS_SERVER_STATUS()

```
#define AGS_SERVER_STATUS(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SERVER_STATUS, AgsServerStatus))
```

AGS_SERVER_STATUS_CLASS()

```
#define AGS_SERVER_STATUS_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_SERVER_STATUS, AgsServerStatusClass))
```

AGS_SERVER_STATUS_GET_CLASS()

```
#define AGS_SERVER_STATUS_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_SERVER_STATUS, AgsServerStatusClass))
```

ags_server_status_get_type ()

```
GType
ags_server_status_get_type ();
```

Types and Values

AGS_TYPE_SERVER_STATUS

```
#define AGS_TYPE_SERVER_STATUS                (ags_server_status_get_type())
```

struct AgsServerStatus

```
struct AgsServerStatus;
```

struct AgsServerStatusClass

```
struct AgsServerStatusClass {
  GObjectClass gobject;
};
```

Chapter 64

AgsServiceProvider

AgsServiceProvider —

Stability Level

Stable, unless otherwise indicated

Functions

gboolean	ags_service_provider_is_operating ()
AgsServerStatus *	ags_service_provider_server_status ()
void	ags_service_provider_set_registry ()
AgsRegistry *	ags_service_provider_get_registry ()
void	ags_service_provider_set_server ()
GList *	ags_service_provider_get_server ()
#define	AGS_IS_SERVICE_PROVIDER()
#define	AGS_IS_SERVICE_PROVIDER_INTERFACE()
#define	AGS_SERVICE_PROVIDER()
#define	AGS_SERVICE_PROVIDER_GET_INTERFACE()
#define	AGS_SERVICE_PROVIDER_INTERFACE()
GType	ags_service_provider_get_type ()

Types and Values

#define	AGS_TYPE_SERVICE_PROVIDER
	AgsServiceProvider
struct	AgsServiceProviderInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsServiceProvider
```

Known Implementations

AgsServiceProvider is implemented by AgsServerApplicationContext.

Description

Functions

ags_service_provider_is_operating ()

```
gboolean
ags_service_provider_is_operating (AgsServiceProvider *service_provider);
```

Check if is operating.

Parameters

service_provider	the AgsServiceProvider

Returns

TRUE if operating, otherwise FALSE

Since: 3.0.0

ags_service_provider_server_status ()

```
AgsServerStatus~*
ags_service_provider_server_status (AgsServiceProvider *service_provider);
```

Get server status.

Parameters

service_provider	the AgsServiceProvider

Returns

the AgsServerStatus

Since: 3.0.0

ags_service_provider_set_registry ()

```
void
ags_service_provider_set_registry (AgsServiceProvider *service_provider,
                                   AgsRegistry *registry);
```

Set registry.

Parameters

service_provider	the AgsServiceProvider	
registry	the AgsRegistry	

Since: 3.0.0

ags_service_provider_get_registry ()

```
AgsRegistry~*
ags_service_provider_get_registry (AgsServiceProvider *service_provider);
```

Get registry.

Parameters

service_provider	the AgsServiceProvider	

Returns

the AgsRegistry

Since: 3.0.0

ags_service_provider_set_server ()

```
void
ags_service_provider_set_server (AgsServiceProvider *service_provider,
                                 GList *server);
```

Set server.

Parameters

service_provider	the AgsServiceProvider	
server	the GList containing AgsServer.	*[element-type Ags.Server]*

Since: 3.0.0

ags_service_provider_get_server ()

```
GList~*
ags_service_provider_get_server (AgsServiceProvider *service_provider);
```

Get server.

Parameters

service_provider	the AgsServiceProvider

Returns

the GList containing AgsServer.

[element-type Ags.Server][transfer full]

Since: 3.0.0

AGS_IS_SERVICE_PROVIDER()

```
#define AGS_IS_SERVICE_PROVIDER(obj)                    (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_SERVICE_PROVIDER))
```

AGS_IS_SERVICE_PROVIDER_INTERFACE()

```
#define AGS_IS_SERVICE_PROVIDER_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_SERVICE_PROVIDER))
```

AGS_SERVICE_PROVIDER()

```
#define AGS_SERVICE_PROVIDER(obj)                       (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SERVICE_PROVIDER, AgsServiceProvider))
```

AGS_SERVICE_PROVIDER_GET_INTERFACE()

```
#define AGS_SERVICE_PROVIDER_GET_INTERFACE(obj)         (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_SERVICE_PROVIDER, AgsServiceProviderInterface))
```

AGS_SERVICE_PROVIDER_INTERFACE()

```
#define AGS_SERVICE_PROVIDER_INTERFACE(vtable)          (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_SERVICE_PROVIDER, AgsServiceProviderInterface))
```

ags_service_provider_get_type ()

```
GType
ags_service_provider_get_type ();
```

Types and Values

AGS_TYPE_SERVICE_PROVIDER

```
#define AGS_TYPE_SERVICE_PROVIDER                       (ags_service_provider_get_type())
```

AgsServiceProvider

```
typedef struct _AgsServiceProvider AgsServiceProvider;
```

struct AgsServiceProviderInterface

```
struct AgsServiceProviderInterface {
  GTypeInterface ginterface;

  gboolean (*is_operating)(AgsServiceProvider *service_provider);

  AgsServerStatus* (*server_status)(AgsServiceProvider *service_provider);

  void (*set_registry)(AgsServiceProvider *service_provider,
           AgsRegistry *registry);
  AgsRegistry* (*get_registry)(AgsServiceProvider *service_provider);

  void (*set_server)(AgsServiceProvider *service_provider,
         GList *server);
  GList* (*get_server)(AgsServiceProvider *service_provider);
};
```

Chapter 65

Controller - providing context path

65.1 AgsController

AgsController — base controller

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CONTROLLER_GET_OBJ_MUTEX()
AgsControllerResource *	ags_controller_resource_alloc ()
void	ags_controller_resource_free ()
void	ags_controller_resource_ref ()
void	ags_controller_resource_unref ()
void	ags_controller_add_resource ()
void	ags_controller_remove_resource ()
AgsControllerResource *	ags_controller_lookup_resource ()
gboolean	ags_controller_query_security_context ()
AgsController *	ags_controller_new ()
#define	AGS_CONTROLLER()
#define	AGS_CONTROLLER_CLASS()
#define	AGS_CONTROLLER_GET_CLASS()
#define	AGS_IS_CONTROLLER()
#define	AGS_IS_CONTROLLER_CLASS()
GType	ags_controller_get_type ()

Properties

char *	context-path	Read / Write
AgsServer *	server	Read / Write

Types and Values

#define	AGS_CONTROLLER_BASE_PATH
struct	AgsControllerResource
#define	AGS_TYPE_CONTROLLER
struct	AgsController
struct	AgsControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsController
        &#x2570;&#x2500;&#x2500; AgsFrontController
```

Includes

```
#include <ags/server/controller/ags_controller.h>
```

Description

The AgsController is a base object to implement controllers.

Functions

AGS_CONTROLLER_GET_OBJ_MUTEX()

```
#define AGS_CONTROLLER_GET_OBJ_MUTEX(obj) (&(((AgsController *) obj)->obj_mutex))
```

ags_controller_resource_alloc ()

```
AgsControllerResource~*
ags_controller_resource_alloc (gchar *group_id,
                               gchar *user_id,
                               guint access_mode);
```

Allocated AgsControllerResource.

Parameters

group_id	the group id	
user_id	the user id	
access_mode	the access mode	

Returns

the newly allocated AgsControllerResource.

[type gpointer][transfer none]

Since: 3.0.0

ags_controller_resource_free ()

```
void
ags_controller_resource_free (AgsControllerResource *controller_resource);
```

Free *controller_resource*.

Parameters

controller_resource	the AgsControllerResource.	*[type gpointer]*

Since: 3.0.0

ags_controller_resource_ref ()

```
void
ags_controller_resource_ref (AgsControllerResource *controller_resource);
```

Increase ref-count of *controller_resource*.

Parameters

controller_resource	the AgsControllerResource.	*[type gpointer]*

Since: 3.0.0

ags_controller_resource_unref ()

```
void
ags_controller_resource_unref (AgsControllerResource *controller_resource);
```

Decrease ref-count of *controller_resource* and free it if ref-count drops to 0.

Parameters

controller_resource	the AgsControllerResource.	*[type gpointer]*

Since: 3.0.0

ags_controller_add_resource ()

```
void
ags_controller_add_resource (AgsController *controller,
                             gchar *resource_name,
                             AgsControllerResource *controller_resource);
```

Add *controller_resource* with key *resource_name* to hash table.

Parameters

controller	the AgsController	
resource_name	the resource name as string	
controller_resource	the AgsControllerResource.	*[type gpointer]*

Since: 3.0.0

ags_controller_remove_resource ()

```
void
ags_controller_remove_resource (AgsController *controller,
                                gchar *resource_name);
```

Remove key *resource_name* from hash table.

Parameters

controller	the AgsController	
resource_name	the resource name as string	

Since: 3.0.0

ags_controller_lookup_resource ()

```
AgsControllerResource~*
ags_controller_lookup_resource (AgsController *controller,
                                gchar *resource_name);
```

Lookup key *resource_name* in hash table.

Parameters

controller	the AgsController	
resource_name	the resource name as string	

Returns

the matching AgsControllerResource.

[type gpointer][transfer none]

Since: 3.0.0

ags_controller_query_security_context ()

```
gboolean
ags_controller_query_security_context (AgsController *controller,
                                       GObject *security_context,
                                       gchar *login);
```

Query *security_context* for *login* .

Parameters

controller	the AgsController	
security_context	the AgsSecurityContext	
login	the login to query	

Returns

TRUE if allowed to proceed, otherwise FALSE

Since: 3.0.0

ags_controller_new ()

```
AgsController~*
ags_controller_new ();
```

Instantiate new AgsController

Returns

the AgsController

Since: 3.0.0

AGS_CONTROLLER()

```
#define AGS_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CONTROLLER, AgsController))
```

AGS_CONTROLLER_CLASS()

```
#define AGS_CONTROLLER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_CONTROLLER, AgsControllerClass))
```

AGS_CONTROLLER_GET_CLASS()

```
#define AGS_CONTROLLER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_CONTROLLER, AgsControllerClass))
```

AGS_IS_CONTROLLER()

```
#define AGS_IS_CONTROLLER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CONTROLLER))
```

AGS_IS_CONTROLLER_CLASS()

```
#define AGS_IS_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CONTROLLER))
```

ags_controller_get_type ()

```
GType
ags_controller_get_type ();
```

Types and Values

AGS_CONTROLLER_BASE_PATH

```
#define AGS_CONTROLLER_BASE_PATH "/ags-xmlrpc"
```

struct AgsControllerResource

```
struct AgsControllerResource {
  gint ref_count;

  gchar *group_id;
  gchar *user_id;

  guint access_mode;
};
```

Specifing attributes of resources.

Members

gint *ref_count*;	the reference count
gchar **group_id*;	the group id of permissions
gchar **user_id*;	the user id of permissions
guint *access_mode*;	the access mode of permissions

AGS_TYPE_CONTROLLER

```
#define AGS_TYPE_CONTROLLER                (ags_controller_get_type())
```

struct AgsController

```
struct AgsController;
```

struct AgsControllerClass

```
struct AgsControllerClass {
  GObjectClass gobject;

  gboolean (*query_security_context)(AgsController *controller,
             GObject *security_context, gchar *login);
};
```

Property Details

The "context-path" property

```
  "context-path"             char~*
```

The context path provided.

Owner: AgsController

Flags: Read / Write

Default value: NULL

Since: 3.0.0

The "server" property

```
  "server"                   AgsServer~*
```

The assigned AgsServer

Owner: AgsController

Flags: Read / Write

Since: 3.0.0

65.2 AgsFrontController

AgsFrontController — handle all XMLRPC requests

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_front_controller_do_request ()
AgsFrontController *	ags_front_controller_new ()
#define	AGS_FRONT_CONTROLLER()
#define	AGS_FRONT_CONTROLLER_CLASS()
#define	AGS_FRONT_CONTROLLER_GET_CLASS()
#define	AGS_IS_FRONT_CONTROLLER()
#define	AGS_IS_FRONT_CONTROLLER_CLASS()
GType	ags_front_controller_get_type ()

Signals

gpointer	do-request	Run Last

Types and Values

#define	AGS_TYPE_FRONT_CONTROLLER
struct	AgsFrontController
struct	AgsFrontControllerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsController
        &#x2570;&#x2500;&#x2500; AgsFrontController
```

Includes

```
#include <ags/server/controller/ags_front_controller.h>
```

Description

The AgsFrontController is an object to handle XMLRPC requests.

Functions

ags_front_controller_do_request ()

```
gpointer
ags_front_controller_do_request (AgsFrontController *front_controller,
                                 SoupServerMessage *msg,
                                 GHashTable *query,
                                 GObject *security_context,
                                 gchar *path,
                                 gchar *login,
                                 gchar *security_token);
```

Do a XML-RPC request for the given *path* with *query* .

Parameters

front_controller	the AgsFrontController	
msg	the SoupServerMessage	
query	the GHashTable	
security_context	the AgsSecurityContext	
path	the context path to access	
login	the login	
security_token	the security token	

Returns

the response on success, otherwise NULL.

Since: 3.0.0

ags_front_controller_new ()

```
AgsFrontController~*
ags_front_controller_new ();
```

Instantiate new AgsFrontController

Returns

the AgsFrontController

Since: 3.0.0

AGS_FRONT_CONTROLLER()

```
#define AGS_FRONT_CONTROLLER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_FRONT_CONTROLLER, AgsFrontController))
```

AGS_FRONT_CONTROLLER_CLASS()

```
#define AGS_FRONT_CONTROLLER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_FRONT_CONTROLLER, AgsFrontControllerClass))
```

AGS_FRONT_CONTROLLER_GET_CLASS()

```
#define AGS_FRONT_CONTROLLER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_FRONT_CONTROLLER, AgsFrontControllerClass))
```

AGS_IS_FRONT_CONTROLLER()

```
#define AGS_IS_FRONT_CONTROLLER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_FRONT_CONTROLLER))
```

AGS_IS_FRONT_CONTROLLER_CLASS()

```
#define AGS_IS_FRONT_CONTROLLER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_FRONT_CONTROLLER))
```

ags_front_controller_get_type ()

```
GType
ags_front_controller_get_type ();
```

Types and Values

AGS_TYPE_FRONT_CONTROLLER

```
#define AGS_TYPE_FRONT_CONTROLLER                 (ags_front_controller_get_type())
```

struct AgsFrontController

```
struct AgsFrontController;
```

struct AgsFrontControllerClass

```
struct AgsFrontControllerClass {
  AgsControllerClass controller;

  gpointer (*do_request)(AgsFrontController *front_controller,
       SoupServerMessage *msg,
       GHashTable *query,
       GObject *security_context,
       gchar *path,
       gchar *login,
       gchar *security_token);
};
```

Signal Details

The "do-request" signal

```
gpointer
user_function (AgsFrontController *front_controller,
               GObject            *msg,
               gpointer            query,
               GObject            *security_context,
               char               *context_path,
               char               *user,
               char               *security_token,
               gpointer            user_data)
```

Do a request on the front controller.

Parameters

front_controller	the AgsFrontController	
msg	the SoupMessage	
query	the GHashTable	
security_context	the AgsSecurityContext	
context_path	the context path to access	
user	the user's UUID	
security_token	the security token	
user_data	user data set when the signal handler was connected.	

Returns

the response

Flags: Run Last

Since: 3.0.0

65.3 AgsPluginController

AgsPluginController — provide abstract controller

Stability Level

Stable, unless otherwise indicated

Functions

gpointer	ags_plugin_controller_do_request ()
#define	AGS_IS_PLUGIN_CONTROLLER()
#define	AGS_IS_PLUGIN_CONTROLLER_INTERFACE()
#define	AGS_PLUGIN_CONTROLLER()
#define	AGS_PLUGIN_CONTROLLER_GET_INTERFACE()
#define	AGS_PLUGIN_CONTROLLER_INTERFACE()
GType	ags_plugin_controller_get_type ()

Types and Values

#define	AGS_TYPE_PLUGIN_CONTROLLER
	AgsPluginController
struct	AgsPluginControllerInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsPluginController
```

Includes

```
#include <ags/server/controller/ags_plugin_controller.h>
```

Description

The AgsPluginController provides you an abstraction of controller implementation.

Functions

ags_plugin_controller_do_request ()

```
gpointer
ags_plugin_controller_do_request (AgsPluginController *plugin_controller,
                                  SoupServerMessage *msg,
                                  GHashTable *query,
                                  GObject *security_context,
                                  gchar *path,
                                  gchar *login,
                                  gchar *security_token);
```

Do request.

Parameters

plugin_controller	the AgsPluginController	
msg	the SoupServerMessage	
query	the GHashTable	
security_context	the AgsSecurityContext	
path	the context path to access	
login	the login	
security_token	the security token	

Returns

the GList containing AgsResponse

Since: 3.0.0

AGS_IS_PLUGIN_CONTROLLER()

```
#define AGS_IS_PLUGIN_CONTROLLER(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_PLUGIN_CONTROLLER))
```

AGS_IS_PLUGIN_CONTROLLER_INTERFACE()

```
#define AGS_IS_PLUGIN_CONTROLLER_INTERFACE(vtable)    (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_PLUGIN_CONTROLLER))
```

AGS_PLUGIN_CONTROLLER()

```
#define AGS_PLUGIN_CONTROLLER(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PLUGIN_CONTROLLER, AgsPluginController))
```

AGS_PLUGIN_CONTROLLER_GET_INTERFACE()

```
#define AGS_PLUGIN_CONTROLLER_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj), ↩
     AGS_TYPE_PLUGIN_CONTROLLER, AgsPluginControllerInterface))
```

AGS_PLUGIN_CONTROLLER_INTERFACE()

```
#define AGS_PLUGIN_CONTROLLER_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_PLUGIN_CONTROLLER, AgsPluginControllerInterface))
```

ags_plugin_controller_get_type ()

```
GType
ags_plugin_controller_get_type ();
```

Types and Values

AGS_TYPE_PLUGIN_CONTROLLER

```
#define AGS_TYPE_PLUGIN_CONTROLLER                    (ags_plugin_controller_get_type())
```

AgsPluginController

```
typedef struct _AgsPluginController AgsPluginController;
```

struct AgsPluginControllerInterface

```
struct AgsPluginControllerInterface {
  GTypeInterface ginterface;

  gpointer (*do_request)(AgsPluginController *plugin_controller,
       SoupServerMessage *msg,
       GHashTable *query,
       GObject *security_context,
       gchar *path,
       gchar *login,
       gchar *security_token);
};
```

Chapter 66

Security - basic authentication

66.1 AgsAuthentication

AgsAuthentication — base password authentication

Stability Level

Stable, unless otherwise indicated

Functions

gchar **	ags_authentication_get_authentication_module ()
gboolean	ags_authentication_login ()
gboolean	ags_authentication_logout ()
gchar *	ags_authentication_generate_token ()
gchar *	ags_authentication_get_digest ()
gboolean	ags_authentication_is_session_active ()
#define	AGS_AUTHENTICATION()
#define	AGS_AUTHENTICATION_GET_INTERFACE()
#define	AGS_AUTHENTICATION_INTERFACE()
#define	AGS_IS_AUTHENTICATION()
#define	AGS_IS_AUTHENTICATION_INTERFACE()
GType	ags_authentication_get_type ()

Types and Values

#define	AGS_TYPE_AUTHENTICATION
	AgsAuthentication
struct	AgsAuthenticationInterface

Object Hierarchy

```
GInterface
&#x2570;&#x2500;&#x2500; AgsAuthentication
```

Known Implementations

AgsAuthentication is implemented by AgsXmlAuthentication.

Includes

```
#include <ags/server/security/ags_authentication.h>
```

Description

The AgsAuthentication interface gives you base authentication functions.

Functions

ags_authentication_get_authentication_module ()

```
gchar~**
ags_authentication_get_authentication_module
                                (AgsAuthentication *authentication);
```

Available authentication modules.

Parameters

authentication	the AgsAuthentication

Returns

a NULL terminated array of strings of available authentication modules.

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

Since: 3.0.0

ags_authentication_login ()

```
gboolean
ags_authentication_login (AgsAuthentication *authentication,
                          gchar *login,
                          gchar *password,
                          gchar **user_uuid,
                          gchar **security_token,
                          GError **error);
```

Login.

Parameters

authentication	the AgsAuthentication	
login	the login	
password	the password	
user_uuid	return location of the user's uuid.	*[out][transfer full]*
security_token	return location of the security token.	*[out][transfer full]*
error	the GError	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_authentication_logout ()

```
gboolean
ags_authentication_logout (AgsAuthentication *authentication,
                           GObject *security_context,
                           gchar *login,
                           gchar *security_token,
                           GError **error);
```

Logout.

Parameters

authentication	the AgsAuthentication	
security_context	the AgsSecurityContext	
login	the login	
security_token	the security token	
error	the GError	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_authentication_generate_token ()

```
gchar~*
ags_authentication_generate_token (AgsAuthentication *authentication,
                                   GError **error);
```

Generate token.

Parameters

authentication	the AgsAuthentication	
error	the GError	

Returns

the generated token.

[transfer full]

Since: 3.0.0

ags_authentication_get_digest ()

```
gchar~*
ags_authentication_get_digest (AgsAuthentication *authentication,
                               gchar *realm,
                               gchar *login,
                               gchar *security_token,
                               GError **error);
```

Get digest of `login`.

Parameters

authentication	the AgsAuthentication	
realm	the realm	
login	the login	
security_token	the security token	
error	the GError	

Returns

the encrypted password.

[transfer full]

Since: 3.0.0

ags_authentication_is_session_active ()

```
gboolean
ags_authentication_is_session_active (AgsAuthentication *authentication,
                                      GObject *security_context,
                                      gchar *user_uuid,
                                      gchar *security_token,
                                      GError **error);
```

Check session.

Parameters

authentication	the AgsAuthentication	
security_context	the AgsSecurityContext	
user_uuid	the user's UUID	

security_token	the security token	
error	the GError	

Returns

TRUE if session active, otherwise FALSE

Since: 3.0.0

AGS_AUTHENTICATION()

```
#define AGS_AUTHENTICATION(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUTHENTICATION, AgsAuthentication))
```

AGS_AUTHENTICATION_GET_INTERFACE()

```
#define AGS_AUTHENTICATION_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_AUTHENTICATION, AgsAuthenticationInterface))
```

AGS_AUTHENTICATION_INTERFACE()

```
#define AGS_AUTHENTICATION_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_AUTHENTICATION, AgsAuthenticationInterface))
```

AGS_IS_AUTHENTICATION()

```
#define AGS_IS_AUTHENTICATION(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_AUTHENTICATION))
```

AGS_IS_AUTHENTICATION_INTERFACE()

```
#define AGS_IS_AUTHENTICATION_INTERFACE(vtable)    (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_AUTHENTICATION))
```

ags_authentication_get_type ()

```
GType
ags_authentication_get_type ();
```

Types and Values

AGS_TYPE_AUTHENTICATION

```
#define AGS_TYPE_AUTHENTICATION                    (ags_authentication_get_type())
```

AgsAuthentication

```
typedef struct _AgsAuthentication AgsAuthentication;
```

struct AgsAuthenticationInterface

```
struct AgsAuthenticationInterface {
  GTypeInterface ginterface;

  gchar** (*get_authentication_module)(AgsAuthentication *authentication);

  gboolean (*login)(AgsAuthentication *authentication,
        gchar *login,
        gchar *password,
        gchar **user_uuid,
        gchar **security_token,
        GError **error);

  gboolean (*logout)(AgsAuthentication *authentication,
         GObject *security_context,
         gchar *login,
         gchar *security_token,
         GError **error);

  gchar* (*generate_token)(AgsAuthentication *authentication,
         GError **error);

  gchar* (*get_digest)(AgsAuthentication *authentication,
           gchar *realm,
           gchar *login,
           gchar *security_token,
           GError **error);

  gboolean (*is_session_active)(AgsAuthentication *authentication,
        GObject *security_context,
        gchar *user_uuid,
        gchar *security_token,
        GError **error);
};
```

66.2 AgsAuthenticationManager

AgsAuthenticationManager — Singleton pattern to organize authentication

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_LOGIN_INFO()
#define	AGS_AUTHENTICATION_MANAGER_GET_OBJ_MUTEX()
AgsLoginInfo *	ags_login_info_alloc ()

void	ags_login_info_free ()
void	ags_login_info_ref ()
void	ags_login_info_unref ()
GList *	ags_authentication_manager_get_authentication ()
void	ags_authentication_manager_add_authentication ()
void	ags_authentication_manager_remove_authentication ()
gint64	ags_authentication_manager_get_session_timeout ()
AgsLoginInfo *	ags_authentication_manager_lookup_login ()
void	ags_authentication_manager_insert_login ()
void	ags_authentication_manager_remove_login ()
gboolean	ags_authentication_manager_login ()
gboolean	ags_authentication_manager_logout ()
gchar *	ags_authentication_manager_get_digest ()
gboolean	ags_authentication_manager_is_session_active ()
AgsAuthenticationManager *	ags_authentication_manager_get_instance ()
AgsAuthenticationManager *	ags_authentication_manager_new ()
#define	AGS_AUTHENTICATION_MANAGER()
#define	AGS_AUTHENTICATION_MANAGER_CLASS()
#define	AGS_AUTHENTICATION_MANAGER_GET_CLASS()
#define	AGS_IS_AUTHENTICATION_MANAGER()
#define	AGS_IS_AUTHENTICATION_MANAGER_CLASS()
GType	ags_authentication_manager_get_type ()

Types and Values

#define	AGS_AUTHENTICATION_MANAGER_DEFAULT_SESSION_TIMEOUT
struct	AgsLoginInfo
#define	AGS_TYPE_AUTHENTICATION_MANAGER
struct	AgsAuthenticationManager
struct	AgsAuthenticationManagerClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsAuthenticationManager
```

Includes

```
#include <ags/server/security/ags_authentication_manager.h>
```

Description

The AgsAuthenticationManager manages your authentication.

Functions

AGS_LOGIN_INFO()

```
#define AGS_LOGIN_INFO(ptr) ((AgsLoginInfo *) ptr)
```

AGS_AUTHENTICATION_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_AUTHENTICATION_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsAuthenticationManager *) obj) ↩
    ->obj_mutex))
```

ags_login_info_alloc ()

```
AgsLoginInfo~*
ags_login_info_alloc ();
```

Alloc AgsLoginInfo.

Returns

the newly allocated AgsLoginInfo.

[type gpointer][transfer none]

Since: 3.0.0

ags_login_info_free ()

```
void
ags_login_info_free (AgsLoginInfo *login_info);
```

Free *login_info* .

Parameters

login_info	the AgsLoginInfo.	*[type gpointer]*

Since: 3.0.0

ags_login_info_ref ()

```
void
ags_login_info_ref (AgsLoginInfo *login_info);
```

Increase ref-count of *login_info* .

Parameters

login_info	the AgsLoginInfo.	*[type gpointer]*

Since: 3.0.0

ags_login_info_unref ()

```
void
ags_login_info_unref (AgsLoginInfo *login_info);
```

Decrease ref-count of *login_info* and free it if ref-count drops to 0.

Parameters

login_info	the AgsLoginInfo.	*[type gpointer]*

Since: 3.0.0

ags_authentication_manager_get_authentication ()

```
GList~*
ags_authentication_manager_get_authentication
                                (AgsAuthenticationManager *authentication_manager);
```

Get authentication.

Parameters

authentication_manager	the AgsAuthenticationManager

Returns

the GList containing GObject implementing AgsAuthentication.

[element-type GObject][transfer full]

Since: 3.0.0

ags_authentication_manager_add_authentication ()

```
void
ags_authentication_manager_add_authentication
                                (AgsAuthenticationManager *authentication_manager,
                                 GObject *authentication);
```

Add *authentication* to *authentication_manager* .

Parameters

authentication_manager	the AgsAuthenticationManager
authentication	the GObject implementing AgsAuthentication

Since: 3.0.0

ags_authentication_manager_remove_authentication ()

```
void
ags_authentication_manager_remove_authentication
                                (AgsAuthenticationManager *authentication_manager,
                                 GObject *authentication);
```

Remove *authentication* from *authentication_manager* .

Parameters

authentication_manager	the AgsAuthenticationManager	
authentication	the GObject implementing AgsAuthentication	

Since: 3.0.0

ags_authentication_manager_get_session_timeout ()

```
gint64
ags_authentication_manager_get_session_timeout
                                (AgsAuthenticationManager *authentication_manager);
```

Get session timeout.

Parameters

authentication_manager	the AgsAuthenticationManager	

Returns

the session timeout

Since: 3.0.0

ags_authentication_manager_lookup_login ()

```
AgsLoginInfo~*
ags_authentication_manager_lookup_login
                                (AgsAuthenticationManager *authentication_manager,
                                 gchar *login);
```

Lookup *login*.

Parameters

authentication_manager	the AgsAuthenticationManager	
login	the login	

Returns

the user AgsLoginInfo or NULL.

[type gpointer][transfer none]

Since: 3.0.0

ags_authentication_manager_insert_login ()

```
void
ags_authentication_manager_insert_login
                                (AgsAuthenticationManager *authentication_manager,
                                 gchar *login,
                                 AgsLoginInfo *login_info);
```

Insert *login* as key and *login_info* as its value.

Parameters

authentication_manager	the AgsAuthenticationManager	
login	the login	
login_info	the AgsLoginInfo	

Since: 3.0.0

ags_authentication_manager_remove_login ()

```
void
ags_authentication_manager_remove_login
                                (AgsAuthenticationManager *authentication_manager,
                                 gchar *login);
```

Remove *login* .

Parameters

authentication_manager	the AgsAuthenticationManager	
login	the login	

Since: 3.0.0

ags_authentication_manager_login ()

```
gboolean
ags_authentication_manager_login (AgsAuthenticationManager *authentication_manager,
                                  gchar *authentication_module,
                                  gchar *login,
                                  gchar *password,
                                  gchar **user_uuid,
                                  gchar **security_token);
```

Login.

Parameters

authentication_manager	the AgsAuthenticationManager	
authentication_module	the authentication module	
login	the login	
password	the password	
user_uuid	return location of user UUID.	*[out][transfer full]*
security_token	return location of security token.	*[out][transfer full]*

Returns

TRUE if login was successful, otherwise FALSE

Since: 3.0.0

ags_authentication_manager_logout ()

```
gboolean
ags_authentication_manager_logout (AgsAuthenticationManager *authentication_manager,
                                   GObject *security_context,
                                   gchar *login,
                                   gchar *security_token);
```

Logout.

Parameters

authentication_manager	the AgsAuthenticationManager	
security_context	the AgsSecurityContext	
login	the login	
security_token	the security token	

Returns

TRUE if logout was successful, otherwise FALSE

Since: 3.0.0

ags_authentication_manager_get_digest ()

```
gchar~*
ags_authentication_manager_get_digest (AgsAuthenticationManager *authentication_manager,
                                       gchar *authentication_module,
                                       gchar *realm,
                                       gchar *login,
                                       gchar *security_token);
```

Get digest of `login`.

Parameters

authentication_manager	the AgsAuthenticationManager	
authentication_module	the authentication module	
realm	the realm	
login	the login	
security_token	the security token	

Returns

the digest as string, or NULL if not available.

[transfer full]

Since: 3.0.0

ags_authentication_manager_is_session_active ()

```
gboolean
ags_authentication_manager_is_session_active
                                (AgsAuthenticationManager *authentication_manager,
                                 GObject *security_context,
                                 gchar *user_uuid,
                                 gchar *security_token);
```

Check if session is active.

Parameters

authentication_manager	the AgsAuthenticationManager	
security_context	the security context	
user_uuid	the user's UUID	
security_token	the security token	

Returns

TRUE if active, otherwise FALSE

Since: 3.0.0

ags_authentication_manager_get_instance ()

```
AgsAuthenticationManager~*
ags_authentication_manager_get_instance
                                ();
```

Get instance.

Returns

the AgsAuthenticationManager.

[transfer none]

Since: 3.0.0

ags_authentication_manager_new ()

```
AgsAuthenticationManager~*
ags_authentication_manager_new ();
```

Creates an AgsAuthenticationManager

Returns

a new AgsAuthenticationManager

Since: 3.0.0

AGS_AUTHENTICATION_MANAGER()

```
#define AGS_AUTHENTICATION_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUTHENTICATION_MANAGER, AgsAuthenticationManager))
```

AGS_AUTHENTICATION_MANAGER_CLASS()

```
#define AGS_AUTHENTICATION_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_AUTHENTICATION_MANAGER, AgsAuthenticationManagerClass))
```

AGS_AUTHENTICATION_MANAGER_GET_CLASS()

```
#define AGS_AUTHENTICATION_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_AUTHENTICATION_MANAGER, AgsAuthenticationManagerClass))
```

AGS_IS_AUTHENTICATION_MANAGER()

```
#define AGS_IS_AUTHENTICATION_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUTHENTICATION_MANAGER))
```

AGS_IS_AUTHENTICATION_MANAGER_CLASS()

```
#define AGS_IS_AUTHENTICATION_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUTHENTICATION_MANAGER))
```

ags_authentication_manager_get_type ()

```
GType
ags_authentication_manager_get_type (void);
```

Types and Values

AGS_AUTHENTICATION_MANAGER_DEFAULT_SESSION_TIMEOUT

```
#define AGS_AUTHENTICATION_MANAGER_DEFAULT_SESSION_TIMEOUT (2 * G_TIME_SPAN_HOUR)
```

struct AgsLoginInfo

```
struct AgsLoginInfo {
  gint ref_count;
  gint active_session_count;

  gchar *user_uuid;

  AgsSecurityContext *security_context;
};
```

AGS_TYPE_AUTHENTICATION_MANAGER

```
#define AGS_TYPE_AUTHENTICATION_MANAGER                (ags_authentication_manager_get_type ↩
    ())
```

struct AgsAuthenticationManager

```
struct AgsAuthenticationManager;
```

struct AgsAuthenticationManagerClass

```
struct AgsAuthenticationManagerClass {
  GObjectClass gobject;
};
```

66.3 AgsBusinessGroup

AgsBusinessGroup — base group permissions

Stability Level

Stable, unless otherwise indicated

Functions

gchar **	ags_business_group_get_group_uuid ()
void	ags_business_group_set_group_name ()
gchar *	ags_business_group_get_group_name ()
void	ags_business_group_set_user ()
gchar **	ags_business_group_get_user ()
#define	AGS_BUSINESS_GROUP()
#define	AGS_BUSINESS_GROUP_GET_INTERFACE()
#define	AGS_BUSINESS_GROUP_INTERFACE()
#define	AGS_IS_BUSINESS_GROUP()
#define	AGS_IS_BUSINESS_GROUP_INTERFACE()
GType	ags_business_group_get_type ()

Types and Values

#define	AGS_TYPE_BUSINESS_GROUP
	AgsBusinessGroup
struct	AgsBusinessGroupInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsBusinessGroup
```

Known Implementations

AgsBusinessGroup is implemented by AgsXmlBusinessGroup.

Includes

```
#include <ags/server/security/ags_business_group.h>
```

Description

The AgsBusinessGroup interface gives you base group permissions.

Functions

ags_business_group_get_group_uuid ()

```
gchar~**
ags_business_group_get_group_uuid (AgsBusinessGroup *business_group,
                                   GObject *security_context,
                                   gchar *user_uuid,
                                   gchar *security_token,
                                   GError **error);
```

Get group UUID as string vector.

Parameters

business_group	the AgsBusinessGroup	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
error	the GError	

Returns

the group UUIDs as NULL terminated string array.

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

Since: 3.0.0

ags_business_group_set_group_name ()

```
void
ags_business_group_set_group_name (AgsBusinessGroup *business_group,
                                   GObject *security_context,
                                   gchar *user_uuid,
                                   gchar *security_token,
                                   gchar *group_uuid,
                                   gchar *group_name,
                                   GError **error);
```

Set business group name.

Parameters

business_group	the AgsBusinessGroup	
security_context	the AgsSecurityContext	
user_uuid	the user's UUID	
security_token	the security token	
group_uuid	the group's UUID	
group_name	the business group's name to set	
error	the GError	

Since: 3.0.0

ags_business_group_get_group_name ()

```
gchar~*
ags_business_group_get_group_name (AgsBusinessGroup *business_group,
                                   GObject *security_context,
                                   gchar *user_uuid,
                                   gchar *security_token,
                                   gchar *group_uuid,
                                   GError **error);
```

Get business group name.

Parameters

business_group	the AgsBusinessGroup	
security_context	the AgsSecurityContext	
user_uuid	the user's UUID	
security_token	the security token	
group_uuid	the group's UUID	
error	the GError	

Returns

the business group's name as string

Since: 3.0.0

ags_business_group_set_user ()

```
void
ags_business_group_set_user (AgsBusinessGroup *business_group,
                             GObject *security_context,
                             gchar *user_uuid,
                             gchar *security_token,
                             gchar *group_uuid,
                             gchar **user,
                             GError **error);
```

Set business group of user names as NULL terminated string array.

Parameters

business_group	the AgsBusinessGroup	
security_context	the AgsSecurityContext	
user_uuid	the user's UUID	
security_token	the security token	
group_uuid	the business group's UUID	
user	the string array containing user names.	*[element-type utf8][array zero-terminated=1]*
error	the GError	

Since: 3.0.0

ags_business_group_get_user ()

```
gchar~**
ags_business_group_get_user (AgsBusinessGroup *business_group,
                             GObject *security_context,
                             gchar *user_uuid,
                             gchar *security_token,
                             gchar *group_uuid,
                             GError **error);
```

Get business group name as NULL terminated string array.

Parameters

business_group	the AgsBusinessGroup	
security_context	the AgsSecurityContext	
user_uuid	the user's UUID	
security_token	the security token	
group_uuid	the business group's UUID	
error	the GError	

Returns

the business group's user names as string vector.

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

Since: 3.0.0

AGS_BUSINESS_GROUP()

```
#define AGS_BUSINESS_GROUP(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_BUSINESS_GROUP, AgsBusinessGroup))
```

AGS_BUSINESS_GROUP_GET_INTERFACE()

```
#define AGS_BUSINESS_GROUP_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_BUSINESS_GROUP, AgsBusinessGroupInterface))
```

AGS_BUSINESS_GROUP_INTERFACE()

```
#define AGS_BUSINESS_GROUP_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_BUSINESS_GROUP, AgsBusinessGroupInterface))
```

AGS_IS_BUSINESS_GROUP()

```
#define AGS_IS_BUSINESS_GROUP(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_BUSINESS_GROUP))
```

AGS_IS_BUSINESS_GROUP_INTERFACE()

```
#define AGS_IS_BUSINESS_GROUP_INTERFACE(vtable)    (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_BUSINESS_GROUP))
```

ags_business_group_get_type ()

```
GType
ags_business_group_get_type ();
```

Types and Values

AGS_TYPE_BUSINESS_GROUP

```
#define AGS_TYPE_BUSINESS_GROUP                    (ags_business_group_get_type())
```

AgsBusinessGroup

```
typedef struct _AgsBusinessGroup AgsBusinessGroup;
```

struct AgsBusinessGroupInterface

```
struct AgsBusinessGroupInterface {
  GTypeInterface ginterface;

  gchar** (*get_group_uuid)(AgsBusinessGroup *business_group,
          GObject *security_context,
          gchar *user_uuid,
          gchar *security_token,
          GError **error);

  void (*set_group_name)(AgsBusinessGroup *business_group,
       GObject *security_context,
       gchar *user_uuid,
       gchar *security_token,
       gchar *group_uuid,
       gchar *group_name,
       GError **error);
  gchar* (*get_group_name)(AgsBusinessGroup *business_group,
         GObject *security_context,
         gchar *user_uuid,
         gchar *security_token,
         gchar *group_uuid,
         GError **error);

  void (*set_user)(AgsBusinessGroup *business_group,
       GObject *security_context,
       gchar *user_uuid,
       gchar *security_token,
       gchar *group_uuid,
       gchar **user,
       GError **error);
  gchar** (*get_user)(AgsBusinessGroup *business_group,
          GObject *security_context,
          gchar *user_uuid,
          gchar *security_token,
          gchar *group_uuid,
          GError **error);
};
```

66.4 AgsBusinessGroupManager

AgsBusinessGroupManager — Singleton pattern to organize business group

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_BUSINESS_GROUP_MANAGER_GET_OBJ_MUTEX()
GList *	ags_business_group_manager_get_business_group ()
void	ags_business_group_manager_add_business_group ()
void	ags_business_group_manager_remove_business_group ()
AgsBusinessGroupManager *	ags_business_group_manager_get_instance ()

AgsBusinessGroupManager *	ags_business_group_manager_new ()
#define	AGS_BUSINESS_GROUP_MANAGER()
#define	AGS_BUSINESS_GROUP_MANAGER_CLASS()
#define	AGS_BUSINESS_GROUP_MANAGER_GET_CLASS()
#define	AGS_IS_BUSINESS_GROUP_MANAGER()
#define	AGS_IS_BUSINESS_GROUP_MANAGER_CLASS()
GType	ags_business_group_manager_get_type ()

Types and Values

#define	AGS_TYPE_BUSINESS_GROUP_MANAGER
struct	AgsBusinessGroupManager
struct	AgsBusinessGroupManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsBusinessGroupManager
```

Includes

```
#include <ags/server/security/ags_business_group_manager.h>
```

Description

The AgsBusinessGroupManager manages your business group.

Functions

AGS_BUSINESS_GROUP_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_BUSINESS_GROUP_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsBusinessGroupManager *) obj) ↩
    ->obj_mutex))
```

ags_business_group_manager_get_business_group ()

```
GList~*
ags_business_group_manager_get_business_group
                                (AgsBusinessGroupManager *business_group_manager);
```

Get business_group.

Parameters

business_group_manager	the AgsBusinessGroupManager

Returns

the GList containing GObject implementing AgsBusinessGroup.

[element-type GObject][transfer full]

Since: 3.0.0

ags_business_group_manager_add_business_group ()

```
void
ags_business_group_manager_add_business_group
                                (AgsBusinessGroupManager *business_group_manager,
                                 GObject *business_group);
```

Add *business_group* to *business_group_manager* .

Parameters

business_group_manager	the AgsBusinessGroupManager	
business_group	the GObject implementing AgsBusinessGroup	

Since: 3.0.0

ags_business_group_manager_remove_business_group ()

```
void
ags_business_group_manager_remove_business_group
                                (AgsBusinessGroupManager *business_group_manager,
                                 GObject *business_group);
```

Remove *business_group* from *business_group_manager* .

Parameters

business_group_manager	the AgsBusinessGroupManager	
business_group	the GObject implementing AgsBusinessGroup	

Since: 3.0.0

ags_business_group_manager_get_instance ()

```
AgsBusinessGroupManager~*
ags_business_group_manager_get_instance
                                ();
```

Get instance.

Returns

the AgsBusinessGroupManager.

[transfer none]

Since: 3.0.0

ags_business_group_manager_new ()

```
AgsBusinessGroupManager~*
ags_business_group_manager_new ();
```

Creates an AgsBusinessGroupManager

Returns

a new AgsBusinessGroupManager

Since: 3.0.0

AGS_BUSINESS_GROUP_MANAGER()

```
#define AGS_BUSINESS_GROUP_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_BUSINESS_GROUP_MANAGER, AgsBusinessGroupManager))
```

AGS_BUSINESS_GROUP_MANAGER_CLASS()

```
#define AGS_BUSINESS_GROUP_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_BUSINESS_GROUP_MANAGER, AgsBusinessGroupManagerClass))
```

AGS_BUSINESS_GROUP_MANAGER_GET_CLASS()

```
#define AGS_BUSINESS_GROUP_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_BUSINESS_GROUP_MANAGER, AgsBusinessGroupManagerClass))
```

AGS_IS_BUSINESS_GROUP_MANAGER()

```
#define AGS_IS_BUSINESS_GROUP_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_BUSINESS_GROUP_MANAGER))
```

AGS_IS_BUSINESS_GROUP_MANAGER_CLASS()

```
#define AGS_IS_BUSINESS_GROUP_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_BUSINESS_GROUP_MANAGER))
```

ags_business_group_manager_get_type ()

```
GType
ags_business_group_manager_get_type (void);
```

Types and Values

AGS_TYPE_BUSINESS_GROUP_MANAGER

```
#define AGS_TYPE_BUSINESS_GROUP_MANAGER                (ags_business_group_manager_get_type ↩
    ())
```

struct AgsBusinessGroupManager

```
struct AgsBusinessGroupManager;
```

struct AgsBusinessGroupManagerClass

```
struct AgsBusinessGroupManagerClass {
  GObjectClass gobject;
};
```

66.5 AgsCertificate

AgsCertificate — base password certificate

Stability Level

Stable, unless otherwise indicated

Functions

gchar **	ags_certificate_get_cert_uuid ()
void	ags_certificate_set_domain ()
gchar *	ags_certificate_get_domain ()
void	ags_certificate_set_key_type ()
gchar *	ags_certificate_get_key_type ()
void	ags_certificate_set_public_key_file ()
gchar *	ags_certificate_get_public_key_file ()
void	ags_certificate_set_private_key_file ()
gchar *	ags_certificate_get_private_key_file ()
#define	AGS_CERTIFICATE()
#define	AGS_CERTIFICATE_GET_INTERFACE()
#define	AGS_CERTIFICATE_INTERFACE()
#define	AGS_IS_CERTIFICATE()
#define	AGS_IS_CERTIFICATE_INTERFACE()
GType	ags_certificate_get_type ()

Types and Values

#define	AGS_TYPE_CERTIFICATE
	AgsCertificate
struct	AgsCertificateInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsCertificate
```

Known Implementations

AgsCertificate is implemented by AgsXmlCertificate.

Includes

```
#include <ags/server/security/ags_certificate.h>
```

Description

The AgsCertificate interface gives you a unique access to all objects and is responsible to set up signal handlers.

Functions

ags_certificate_get_cert_uuid ()

```
gchar~**
ags_certificate_get_cert_uuid (AgsCertificate *certificate,
                               GObject *security_context,
                               gchar *user_uuid,
                               gchar *security_token,
                               GError **error);
```

Get cert UUID as string vector.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
error	the GError	

Returns

the cert UUIDs as NULL terminated string array.

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

Since: 3.0.0

ags_certificate_set_domain ()

```
void
ags_certificate_set_domain (AgsCertificate *certificate,
                            GObject *security_context,
                            gchar *user_uuid,
                            gchar *security_token,
                            gchar *cert_uuid,
                            gchar *domain,
                            GError **error);
```

Set domain.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
domain	the domain	
error	the GError	

Since: 3.0.0

ags_certificate_get_domain ()

```
gchar~*
ags_certificate_get_domain (AgsCertificate *certificate,
                            GObject *security_context,
                            gchar *user_uuid,
                            gchar *security_token,
                            gchar *cert_uuid,
                            GError **error);
```

Get domain.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
error	the GError	

Returns

the domain.

[transfer full]

Since: 3.0.0

ags_certificate_set_key_type ()

```
void
ags_certificate_set_key_type (AgsCertificate *certificate,
                              GObject *security_context,
                              gchar *user_uuid,
                              gchar *security_token,
                              gchar *cert_uuid,
                              gchar *key_type,
                              GError **error);
```

Set key type.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
key_type	the key type	
error	the GError	

Since: 3.0.0

ags_certificate_get_key_type ()

```
gchar~*
ags_certificate_get_key_type (AgsCertificate *certificate,
                              GObject *security_context,
                              gchar *user_uuid,
                              gchar *security_token,
                              gchar *cert_uuid,
                              GError **error);
```

Get key type.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
error	the GError	

Returns

the key type.

[transfer full]

Since: 3.0.0

ags_certificate_set_public_key_file ()

```
void
ags_certificate_set_public_key_file (AgsCertificate *certificate,
                                     GObject *security_context,
                                     gchar *user_uuid,
                                     gchar *security_token,
                                     gchar *cert_uuid,
                                     gchar *public_key_file,
                                     GError **error);
```

Set public key file.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
public_key_file	the public key file	
error	the GError	

Since: 3.0.0

ags_certificate_get_public_key_file ()

```
gchar~*
ags_certificate_get_public_key_file (AgsCertificate *certificate,
                                     GObject *security_context,
                                     gchar *user_uuid,
                                     gchar *security_token,
                                     gchar *cert_uuid,
                                     GError **error);
```

Get public key file.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
error	the GError	

Returns

the public key file.

[transfer full]

Since: 3.0.0

ags_certificate_set_private_key_file ()

```
void
ags_certificate_set_private_key_file (AgsCertificate *certificate,
                                      GObject *security_context,
                                      gchar *user_uuid,
                                      gchar *security_token,
                                      gchar *cert_uuid,
                                      gchar *private_key_file,
                                      GError **error);
```

Set private key file.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
private_key_file	the private key file	
error	the GError	

Since: 3.0.0

ags_certificate_get_private_key_file ()

```
gchar~*
ags_certificate_get_private_key_file (AgsCertificate *certificate,
                                      GObject *security_context,
                                      gchar *user_uuid,
                                      gchar *security_token,
                                      gchar *cert_uuid,
                                      GError **error);
```

Get private key file.

Parameters

certificate	the AgsCertificate	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
cert_uuid	the cert's UUID	
error	the GError	

Returns

the private key file.

[transfer full]

Since: 3.0.0

AGS_CERTIFICATE()

```
#define AGS_CERTIFICATE(obj)                    (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CERTIFICATE, AgsCertificate))
```

AGS_CERTIFICATE_GET_INTERFACE()

```
#define AGS_CERTIFICATE_GET_INTERFACE(obj)      (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_CERTIFICATE, AgsCertificateInterface))
```

AGS_CERTIFICATE_INTERFACE()

```
#define AGS_CERTIFICATE_INTERFACE(vtable)       (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_CERTIFICATE, AgsCertificateInterface))
```

AGS_IS_CERTIFICATE()

```
#define AGS_IS_CERTIFICATE(obj)                 (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_CERTIFICATE))
```

AGS_IS_CERTIFICATE_INTERFACE()

```
#define AGS_IS_CERTIFICATE_INTERFACE(vtable)    (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_CERTIFICATE))
```

ags_certificate_get_type ()

```
GType
ags_certificate_get_type ();
```

Types and Values

AGS_TYPE_CERTIFICATE

```
#define AGS_TYPE_CERTIFICATE                    (ags_certificate_get_type())
```

AgsCertificate

```
typedef struct _AgsCertificate AgsCertificate;
```

struct AgsCertificateInterface

```
struct AgsCertificateInterface {
  GTypeInterface ginterface;

  gchar** (*get_cert_uuid)(AgsCertificate *certificate,
         GObject *security_context,
         gchar *user_uuid,
         gchar *security_token,
         GError **error);

  void (*set_domain)(AgsCertificate *certificate,
         GObject *security_context,
         gchar *user_uuid,
         gchar *security_token,
         gchar *cert_uuid,
         gchar *domain,
         GError **error);
  gchar* (*get_domain)(AgsCertificate *certificate,
           GObject *security_context,
           gchar *user_uuid,
           gchar *security_token,
           gchar *cert_uuid,
           GError **error);

  void (*set_key_type)(AgsCertificate *certificate,
           GObject *security_context,
           gchar *user_uuid,
           gchar *security_token,
           gchar *cert_uuid,
           gchar *key_type,
           GError **error);
  gchar* (*get_key_type)(AgsCertificate *certificate,
       GObject *security_context,
       gchar *user_uuid,
       gchar *security_token,
       gchar *cert_uuid,
       GError **error);

  void (*set_public_key_file)(AgsCertificate *certificate,
            GObject *security_context,
            gchar *user_uuid,
            gchar *security_token,
            gchar *cert_uuid,
            gchar *public_key_file,
            GError **error);
  gchar* (*get_public_key_file)(AgsCertificate *certificate,
        GObject *security_context,
        gchar *user_uuid,
        gchar *security_token,
        gchar *cert_uuid,
        GError **error);

  void (*set_private_key_file)(AgsCertificate *certificate,
             GObject *security_context,
             gchar *user_uuid,
             gchar *security_token,
             gchar *cert_uuid,
             gchar *private_key_file,
             GError **error);
  gchar* (*get_private_key_file)(AgsCertificate *certificate,
         GObject *security_context,
```

```
        gchar *user_uuid,
        gchar *security_token,
        gchar *cert_uuid,
        GError **error);
};
```

66.6 AgsCertificateManager

AgsCertificateManager — Singleton pattern to organize certificate

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_CERTIFICATE_MANAGER_GET_OBJ_MUTEX()
GList *	ags_certificate_manager_get_certificate ()
void	ags_certificate_manager_add_certificate ()
void	ags_certificate_manager_remove_certificate ()
void	ags_certificate_manager_get_key_file ()
AgsCertificateManager *	ags_certificate_manager_get_instance ()
AgsCertificateManager *	ags_certificate_manager_new ()
#define	AGS_CERTIFICATE_MANAGER()
#define	AGS_CERTIFICATE_MANAGER_CLASS()
#define	AGS_CERTIFICATE_MANAGER_GET_CLASS()
#define	AGS_IS_CERTIFICATE_MANAGER()
#define	AGS_IS_CERTIFICATE_MANAGER_CLASS()
GType	ags_certificate_manager_get_type ()

Types and Values

#define	AGS_TYPE_CERTIFICATE_MANAGER
struct	AgsCertificateManager
struct	AgsCertificateManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsCertificateManager
```

Includes

```
#include <ags/server/security/ags_certificate_manager.h>
```

Description

The AgsCertificateManager manages your certificate.

Functions

AGS_CERTIFICATE_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_CERTIFICATE_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsCertificateManager *) obj)-> ↩
    obj_mutex))
```

ags_certificate_manager_get_certificate ()

```
GList~*
ags_certificate_manager_get_certificate
                                (AgsCertificateManager *certificate_manager);
```

Get certificate.

Parameters

certificate_manager	the AgsCertificateManager

Returns

the GList containing GObject implementing AgsCertificate.

[element-type GObject][transfer full]

Since: 3.0.0

ags_certificate_manager_add_certificate ()

```
void
ags_certificate_manager_add_certificate
                                (AgsCertificateManager *certificate_manager,
                                 GObject *certificate);
```

Add *certificate* to *certificate_manager* .

Parameters

certificate_manager	the AgsCertificateManager
certificate	the GObject implementing AgsCertificate

Since: 3.0.0

ags_certificate_manager_remove_certificate ()

```
void
ags_certificate_manager_remove_certificate
                                (AgsCertificateManager *certificate_manager,
                                 GObject *certificate);
```

Remove *certificate* from *certificate_manager* .

Parameters

certificate_manager	the AgsCertificateManager	
certificate	the GObject implementing AgsCertificate	

Since: 3.0.0

ags_certificate_manager_get_key_file ()

```
void
ags_certificate_manager_get_key_file (AgsCertificateManager *certificate_manager,
                                      GObject *security_context,
                                      gchar *user_uuid,
                                      gchar *security_token,
                                      gchar *domain,
                                      gchar *key_type,
                                      gchar **public_key_file,
                                      gchar **private_key_file);
```

Get public and private key file.

Parameters

certificate_manager	the AgsCertificateManager	
security_context	the AgsSecurityContext	
user_uuid	the user's UUID	
security_token	the security token	
domain	the domain to find	
key_type	the key type to find	
public_key_file	return location of public key filename.	*[out][transfer full]*
private_key_file	return location of private key filename.	*[out][transfer full]*

Since: 3.0.0

ags_certificate_manager_get_instance ()

```
AgsCertificateManager~*
ags_certificate_manager_get_instance ();
```

Get instance.

Returns

the AgsCertificateManager.

[transfer none]

Since: 3.0.0

ags_certificate_manager_new ()

```
AgsCertificateManager~*
ags_certificate_manager_new ();
```

Creates an AgsCertificateManager

Returns

a new AgsCertificateManager

Since: 3.0.0

AGS_CERTIFICATE_MANAGER()

```
#define AGS_CERTIFICATE_MANAGER(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_CERTIFICATE_MANAGER, AgsCertificateManager))
```

AGS_CERTIFICATE_MANAGER_CLASS()

```
#define AGS_CERTIFICATE_MANAGER_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_CERTIFICATE_MANAGER, AgsCertificateManagerClass))
```

AGS_CERTIFICATE_MANAGER_GET_CLASS()

```
#define AGS_CERTIFICATE_MANAGER_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_CERTIFICATE_MANAGER, AgsCertificateManagerClass))
```

AGS_IS_CERTIFICATE_MANAGER()

```
#define AGS_IS_CERTIFICATE_MANAGER(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_CERTIFICATE_MANAGER))
```

AGS_IS_CERTIFICATE_MANAGER_CLASS()

```
#define AGS_IS_CERTIFICATE_MANAGER_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_CERTIFICATE_MANAGER))
```

ags_certificate_manager_get_type ()

```
GType
ags_certificate_manager_get_type (void);
```

Types and Values

AGS_TYPE_CERTIFICATE_MANAGER

```
#define AGS_TYPE_CERTIFICATE_MANAGER                (ags_certificate_manager_get_type())
```

struct AgsCertificateManager

```
struct AgsCertificateManager;
```

struct AgsCertificateManagerClass

```
struct AgsCertificateManagerClass {
  GObjectClass gobject;
};
```

66.7 AgsPasswordStore

AgsPasswordStore — password storage

Stability Level

Stable, unless otherwise indicated

Functions

void	ags_password_store_set_login_name ()
gchar *	ags_password_store_get_login_name ()
void	ags_password_store_set_password ()
gchar *	ags_password_store_get_password ()
gchar *	ags_password_store_encrypt_password ()
#define	AGS_IS_PASSWORD_STORE()
#define	AGS_IS_PASSWORD_STORE_INTERFACE()
#define	AGS_PASSWORD_STORE()
#define	AGS_PASSWORD_STORE_GET_INTERFACE()
#define	AGS_PASSWORD_STORE_INTERFACE()
GType	ags_password_store_get_type ()

Types and Values

#define	AGS_TYPE_PASSWORD_STORE
	AgsPasswordStore
struct	AgsPasswordStoreInterface

Object Hierarchy

```
    GInterface
    &#x2570;&#x2500;&#x2500; AgsPasswordStore
```

Known Implementations

AgsPasswordStore is implemented by AgsXmlPasswordStore.

Includes

```
#include <ags/server/security/ags_password_store.h>
```

Description

The AgsPasswordStore interface gives you password storage.

Functions

ags_password_store_set_login_name ()

```
void
ags_password_store_set_login_name (AgsPasswordStore *password_store,
                                   GObject *security_context,
                                   gchar *user_uuid,
                                   gchar *security_token,
                                   gchar *login_name,
                                   GError **error);
```

Set login name.

Parameters

password_store	the AgsPasswordStore	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
login_name	the login name	
error	the GError	

Since: 3.0.0

ags_password_store_get_login_name ()

```
gchar~*
ags_password_store_get_login_name (AgsPasswordStore *password_store,
                                   GObject *security_context,
                                   gchar *user_uuid,
                                   gchar *security_token,
                                   GError **error);
```

Get login name.

Parameters

password_store	the AgsPasswordStore	
security_context	the AgsSecurityContext	
user_uuid	the users unique identifier	
security_token	the security token	
error	the GError	

Returns

the login name.

[transfer full]

Since: 3.0.0

ags_password_store_set_password ()

```
void
ags_password_store_set_password (AgsPasswordStore *password_store,
                                 GObject *security_context,
                                 gchar *user_uuid,
                                 gchar *security_token,
                                 gchar *password,
                                 GError **error);
```

Set password.

Parameters

password_store	the AgsPasswordStore	
security_context	the AgsSecurityContext	
user_uuid	the user UUID	
security_token	the security token	
password	the password	
error	the GError	

Since: 3.0.0

ags_password_store_get_password ()

```
gchar~*
ags_password_store_get_password (AgsPasswordStore *password_store,
                                 GObject *security_context,
                                 gchar *user_uuid,
                                 gchar *security_token,
                                 GError **error);
```

Get password.

Parameters

password_store	the AgsPasswordStore	
security_context	the AgsSecurityContext	
user_uuid	the user UUID	
security_token	the security token	
error	the GError	

Returns

the password.

[transfer full]

Since: 3.0.0

ags_password_store_encrypt_password ()

```
gchar~*
ags_password_store_encrypt_password (AgsPasswordStore *password_store,
                                     gchar *password,
                                     gchar *salt,
                                     GError **error);
```

Encrypt password.

Parameters

password_store	the AgsPasswordStore	
password	the password	
salt	your salt	
error	the GError	

Returns

the encrypted bytes.

[transfer full]

Since: 3.0.0

AGS_IS_PASSWORD_STORE()

```
#define AGS_IS_PASSWORD_STORE(obj)                  (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    AGS_TYPE_PASSWORD_STORE))
```

AGS_IS_PASSWORD_STORE_INTERFACE()

```
#define AGS_IS_PASSWORD_STORE_INTERFACE(vtable)     (G_TYPE_CHECK_CLASS_TYPE((vtable),  ↩
    AGS_TYPE_PASSWORD_STORE))
```

AGS_PASSWORD_STORE()

```
#define AGS_PASSWORD_STORE(obj)                     (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PASSWORD_STORE, AgsPasswordStore))
```

AGS_PASSWORD_STORE_GET_INTERFACE()

```
#define AGS_PASSWORD_STORE_GET_INTERFACE(obj)       (G_TYPE_INSTANCE_GET_INTERFACE((obj),  ↩
    AGS_TYPE_PASSWORD_STORE, AgsPasswordStoreInterface))
```

AGS_PASSWORD_STORE_INTERFACE()

```
#define AGS_PASSWORD_STORE_INTERFACE(vtable)        (G_TYPE_CHECK_CLASS_CAST((vtable),  ↩
    AGS_TYPE_PASSWORD_STORE, AgsPasswordStoreInterface))
```

ags_password_store_get_type ()

```
GType
ags_password_store_get_type ();
```

Types and Values

AGS_TYPE_PASSWORD_STORE

```
#define AGS_TYPE_PASSWORD_STORE                    (ags_password_store_get_type())
```

AgsPasswordStore

```
typedef struct _AgsPasswordStore AgsPasswordStore;
```

struct AgsPasswordStoreInterface

```
struct AgsPasswordStoreInterface {
  GTypeInterface ginterface;

  void (*set_login_name)(AgsPasswordStore *password_store,
       GObject *security_context,
       gchar *user_uuid,
       gchar *security_token,
       gchar *login_name,
       GError **error);
  gchar* (*get_login_name)(AgsPasswordStore *password_store,
         GObject *security_context,
         gchar *user_uuid,
         gchar *security_token,
         GError **error);

  void (*set_password)(AgsPasswordStore *password_store,
           GObject *security_context,
           gchar *user_uuid,
           gchar *security_token,
           gchar *password,
           GError **error);
  gchar* (*get_password)(AgsPasswordStore *password_store,
       GObject *security_context,
       gchar *user_uuid,
       gchar *security_token,
       GError **error);

  gchar* (*encrypt_password)(AgsPasswordStore *password_store,
           gchar *password,
           gchar *salt,
           GError **error);
};
```

66.8 AgsPasswordStoreManager

AgsPasswordStoreManager — Singleton pattern to organize password stores

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_PASSWORD_STORE_MANAGER_GET_OBJ_MUTEX()
GList *	ags_password_store_manager_get_password_store ()
void	ags_password_store_manager_add_password_store ()
void	ags_password_store_manager_remove_password_store ()
gboolean	ags_password_store_manager_check_password ()
AgsPasswordStoreManager *	ags_password_store_manager_get_instance ()
AgsPasswordStoreManager *	ags_password_store_manager_new ()
#define	AGS_IS_PASSWORD_STORE_MANAGER()
#define	AGS_IS_PASSWORD_STORE_MANAGER_CLASS()
#define	AGS_PASSWORD_STORE_MANAGER()
#define	AGS_PASSWORD_STORE_MANAGER_CLASS()
#define	AGS_PASSWORD_STORE_MANAGER_GET_CLASS()
GType	ags_password_store_manager_get_type ()

Types and Values

#define	AGS_TYPE_PASSWORD_STORE_MANAGER
struct	AgsPasswordStoreManager
struct	AgsPasswordStoreManagerClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsPasswordStoreManager
```

Includes

```
#include <ags/server/security/ags_password_store_manager.h>
```

Description

The AgsPasswordStoreManager manages your password stores.

Functions

AGS_PASSWORD_STORE_MANAGER_GET_OBJ_MUTEX()

```
#define AGS_PASSWORD_STORE_MANAGER_GET_OBJ_MUTEX(obj) (&(((AgsPasswordStoreManager *) obj) ↩
    ->obj_mutex))
```

ags_password_store_manager_get_password_store ()

```
GList~*
ags_password_store_manager_get_password_store
                                (AgsPasswordStoreManager *password_store_manager);
```

Get password_store.

Parameters

password_store_manager	the AgsPasswordStoreManager

Returns

the GList containing GObject implementing AgsPasswordStore.

[element-type GObject][transfer full]

Since: 3.0.0

ags_password_store_manager_add_password_store ()

```
void
ags_password_store_manager_add_password_store
                                (AgsPasswordStoreManager *password_store_manager,
                                 GObject *password_store);
```

Add *password_store* to *password_store_manager* .

Parameters

password_store_manager	the AgsPasswordStoreManager
password_store	the GObject implementing AgsPasswordStore

Since: 3.0.0

ags_password_store_manager_remove_password_store ()

```
void
ags_password_store_manager_remove_password_store
                                (AgsPasswordStoreManager *password_store_manager,
                                 GObject *password_store);
```

Remove *password_store* from *password_store_manager* .

Parameters

password_store_manager	the AgsPasswordStoreManager	
password_store	the GObject implementing AgsPasswordStore	

Since: 3.0.0

ags_password_store_manager_check_password ()

```
gboolean
ags_password_store_manager_check_password
                                (AgsPasswordStoreManager *password_store_manager,
                                 gchar *user_uuid,
                                 gchar *password);
```

Check *password* to be valid for *user_uuid* .

Parameters

password_store_manager	the AgsPasswordStoreManager	
user_uuid	the user uuid	
password	the password	

Returns

TRUE if password matches, otherwise FALSE

Since: 3.0.0

ags_password_store_manager_get_instance ()

```
AgsPasswordStoreManager~*
ags_password_store_manager_get_instance
                                ();
```

Get instance.

Returns

the AgsPasswordStoreManager.

[transfer none]

Since: 3.0.0

ags_password_store_manager_new ()

```
AgsPasswordStoreManager~*
ags_password_store_manager_new ();
```

Creates an AgsPasswordStoreManager

Returns

a new AgsPasswordStoreManager

Since: 3.0.0

AGS_IS_PASSWORD_STORE_MANAGER()

```
#define AGS_IS_PASSWORD_STORE_MANAGER(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_PASSWORD_STORE_MANAGER))
```

AGS_IS_PASSWORD_STORE_MANAGER_CLASS()

```
#define AGS_IS_PASSWORD_STORE_MANAGER_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_PASSWORD_STORE_MANAGER))
```

AGS_PASSWORD_STORE_MANAGER()

```
#define AGS_PASSWORD_STORE_MANAGER(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_PASSWORD_STORE_MANAGER, AgsPasswordStoreManager))
```

AGS_PASSWORD_STORE_MANAGER_CLASS()

```
#define AGS_PASSWORD_STORE_MANAGER_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST((class),  ↩
    AGS_TYPE_PASSWORD_STORE_MANAGER, AgsPasswordStoreManagerClass))
```

AGS_PASSWORD_STORE_MANAGER_GET_CLASS()

```
#define AGS_PASSWORD_STORE_MANAGER_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS ((obj),  ↩
    AGS_TYPE_PASSWORD_STORE_MANAGER, AgsPasswordStoreManagerClass))
```

ags_password_store_manager_get_type ()

```
GType
ags_password_store_manager_get_type (void);
```

Types and Values

AGS_TYPE_PASSWORD_STORE_MANAGER

```
#define AGS_TYPE_PASSWORD_STORE_MANAGER                 (ags_password_store_manager_get_type ↩
    ())
```

struct AgsPasswordStoreManager

```
struct AgsPasswordStoreManager;
```

struct AgsPasswordStoreManagerClass

```
struct AgsPasswordStoreManagerClass {
  GObjectClass gobject;
};
```

66.9 AgsAuthSecurityContext

AgsAuthSecurityContext — auth security context

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_AUTH_SECURITY_CONTEXT_GET_OBJ_MUTEX()
AgsAuthSecurityContext *	ags_auth_security_context_get_instance ()
AgsAuthSecurityContext *	ags_auth_security_context_new ()
#define	AGS_AUTH_SECURITY_CONTEXT()
#define	AGS_AUTH_SECURITY_CONTEXT_CLASS()
#define	AGS_AUTH_SECURITY_CONTEXT_GET_CLASS()
#define	AGS_IS_AUTH_SECURITY_CONTEXT()
#define	AGS_IS_AUTH_SECURITY_CONTEXT_CLASS()
GType	ags_auth_security_context_get_type ()

Types and Values

#define	AGS_TYPE_AUTH_SECURITY_CONTEXT
struct	AgsAuthSecurityContext
struct	AgsAuthSecurityContextClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsSecurityContext
        &#x2570;&#x2500;&#x2500; AgsAuthSecurityContext
```

Includes

```
#include <ags/server/security/ags_auth_security_context.h>
```

Description

The AgsAuthSecurityContext has got all available privileges to do anything.

Functions

AGS_AUTH_SECURITY_CONTEXT_GET_OBJ_MUTEX()

```
#define AGS_AUTH_SECURITY_CONTEXT_GET_OBJ_MUTEX(obj) (&(((AgsAuthSecurityContext *) obj)-> ↩
    obj_mutex))
```

ags_auth_security_context_get_instance ()

```
AgsAuthSecurityContext~*
ags_auth_security_context_get_instance
                                ();
```

Get instance.

Returns

the AgsAuthSecurityContext.

[transfer none]

Since: 3.0.0

ags_auth_security_context_new ()

```
AgsAuthSecurityContext~*
ags_auth_security_context_new ();
```

Create AgsAuthSecurityContext.

Returns

the new AgsAuthSecurityContext instance

Since: 3.0.0

AGS_AUTH_SECURITY_CONTEXT()

```
#define AGS_AUTH_SECURITY_CONTEXT(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_AUTH_SECURITY_CONTEXT, AgsAuthSecurityContext))
```

AGS_AUTH_SECURITY_CONTEXT_CLASS()

```
#define AGS_AUTH_SECURITY_CONTEXT_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_AUTH_SECURITY_CONTEXT, AgsAuthSecurityContextClass))
```

AGS_AUTH_SECURITY_CONTEXT_GET_CLASS()

```
#define AGS_AUTH_SECURITY_CONTEXT_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_AUTH_SECURITY_CONTEXT, AgsAuthSecurityContextClass))
```

AGS_IS_AUTH_SECURITY_CONTEXT()

```
#define AGS_IS_AUTH_SECURITY_CONTEXT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_AUTH_SECURITY_CONTEXT))
```

AGS_IS_AUTH_SECURITY_CONTEXT_CLASS()

```
#define AGS_IS_AUTH_SECURITY_CONTEXT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_AUTH_SECURITY_CONTEXT))
```

ags_auth_security_context_get_type ()

```
GType
ags_auth_security_context_get_type ();
```

Types and Values

AGS_TYPE_AUTH_SECURITY_CONTEXT

```
#define AGS_TYPE_AUTH_SECURITY_CONTEXT                 (ags_auth_security_context_get_type() ↩
    )
```

struct AgsAuthSecurityContext

```
struct AgsAuthSecurityContext;
```

struct AgsAuthSecurityContextClass

```
struct AgsAuthSecurityContextClass {
  AgsSecurityContextClass security_context;
};
```

66.10 AgsSecurityContext

AgsSecurityContext — current security context

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_SECURITY_CONTEXT_GET_OBJ_MUTEX()
void	ags_security_context_parse_business_group ()
gchar **	ags_security_context_get_business_group ()
void	ags_security_context_add_server_context ()
gboolean	ags_security_context_remove_server_context ()
gchar **	ags_security_context_get_server_context ()
AgsSecurityContext *	ags_security_context_new ()
#define	AGS_IS_SECURITY_CONTEXT()
#define	AGS_IS_SECURITY_CONTEXT_CLASS()
#define	AGS_SECURITY_CONTEXT()
#define	AGS_SECURITY_CONTEXT_CLASS()
#define	AGS_SECURITY_CONTEXT_GET_CLASS()
GType	ags_security_context_get_type ()
GType	ags_security_context_mode_get_type ()

Properties

char *	certs	Read / Write

Types and Values

enum	AgsSecurityContextMode
#define	AGS_TYPE_SECURITY_CONTEXT
#define	AGS_TYPE_SECURITY_CONTEXT_MODE
struct	AgsSecurityContext
struct	AgsSecurityContextClass

Object Hierarchy

```
    GFlags
    &#x2570;&#x2500;&#x2500; AgsSecurityContextMode
    GObject
    &#x2570;&#x2500;&#x2500; AgsSecurityContext
        &#x2570;&#x2500;&#x2500; AgsAuthSecurityContext
```

Includes

```
#include <ags/server/security/ags_security_context.h>
```

Description

The AgsSecurityContext is an object to track active server contices.

Functions

AGS_SECURITY_CONTEXT_GET_OBJ_MUTEX()

```
#define AGS_SECURITY_CONTEXT_GET_OBJ_MUTEX(obj) (&(((AgsSecurityContext *) obj)->obj_mutex) ↩
    )
```

ags_security_context_parse_business_group ()

```
void
ags_security_context_parse_business_group
                                (AgsSecurityContext *security_context,
                                 xmlDoc *xml_doc,
                                 gchar *user_uuid);
```

Parse *business_group* and apply to *security_context* .

Parameters

security_context	the AgsSecurityContext	
xml_doc	the xmlDoc containing groups	
user_uuid	the user's UUID	

Since: 3.0.0

ags_security_context_get_business_group ()

```
gchar~**
ags_security_context_get_business_group
                                (AgsSecurityContext *security_context);
```

Get business group of *security_context* .

Parameters

security_context	the AgsSecurityContext	

Returns

the string vector containing business group or NULL.

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

Since: 3.0.0

ags_security_context_add_server_context ()

```
void
ags_security_context_add_server_context
                                (AgsSecurityContext *security_context,
                                 gchar *server_context);
```

Add *server_context* to *security_context* .

Parameters

security_context	the AgsSecurityContext	
server_context	the server context	

Since: 3.0.0

ags_security_context_remove_server_context ()

```
gboolean
ags_security_context_remove_server_context
                                (AgsSecurityContext *security_context,
                                 gchar *server_context);
```

Remove *server_context* from *security_context* .

Parameters

security_context	the AgsSecurityContext	
server_context	the server context	

Returns

TRUE on success, otherwise FALSE

Since: 3.0.0

ags_security_context_get_server_context ()

```
gchar~**
ags_security_context_get_server_context
                                (AgsSecurityContext *security_context);
```

Get business group of *security_context* .

Parameters

security_context	the AgsSecurityContext	

Returns

the string vector containing business group or NULL.

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

Since: 3.0.0

ags_security_context_new ()

```
AgsSecurityContext~*
ags_security_context_new ();
```

Create AgsSecurityContext.

Returns

the new AgsSecurityContext instance

Since: 3.0.0

AGS_IS_SECURITY_CONTEXT()

```
#define AGS_IS_SECURITY_CONTEXT(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_SECURITY_CONTEXT))
```

AGS_IS_SECURITY_CONTEXT_CLASS()

```
#define AGS_IS_SECURITY_CONTEXT_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_SECURITY_CONTEXT))
```

AGS_SECURITY_CONTEXT()

```
#define AGS_SECURITY_CONTEXT(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_SECURITY_CONTEXT, AgsSecurityContext))
```

AGS_SECURITY_CONTEXT_CLASS()

```
#define AGS_SECURITY_CONTEXT_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_SECURITY_CONTEXT, AgsSecurityContextClass))
```

AGS_SECURITY_CONTEXT_GET_CLASS()

```
#define AGS_SECURITY_CONTEXT_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_SECURITY_CONTEXT, AgsSecurityContextClass))
```

ags_security_context_get_type ()

```
GType
ags_security_context_get_type ();
```

ags_security_context_mode_get_type ()

```
GType
ags_security_context_mode_get_type ();
```

Types and Values

enum AgsSecurityContextMode

The AgsSecurityContextMode specifies basic account permissions.

Members

AGS_SECURITY_CONTEXT_RPC_READ	allowed to read using RPC
AGS_SECURITY_CONTEXT_RPC_WRITE	allowed to write using RPC
AGS_SECURITY_CONTEXT_RPC_EXECUTE	allowed to execute using RPC
AGS_SECURITY_CONTEXT_ACCOUNT_READ	allowed to read account information
AGS_SECURITY_CONTEXT_ACCOUNT_WRITE	allowed to write account information

AGS_TYPE_SECURITY_CONTEXT

```
#define AGS_TYPE_SECURITY_CONTEXT                (ags_security_context_get_type())
```

AGS_TYPE_SECURITY_CONTEXT_MODE

```
#define AGS_TYPE_SECURITY_CONTEXT_MODE           (ags_security_context_mode_get_type())
```

struct AgsSecurityContext

```
struct AgsSecurityContext;
```

struct AgsSecurityContextClass

```
struct AgsSecurityContextClass {
  GObjectClass gobject;
};
```

Property Details

The "certs" property

```
  "certs"                    char~*
```

The assigned certificates as string.

Owner: AgsSecurityContext

Flags: Read / Write

Default value: NULL

Since: 3.0.0

66.11 AgsXmlAuthentication

AgsXmlAuthentication — authentication by XML file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_XML_AUTHENTICATION_GET_OBJ_MUTEX()
void	ags_xml_authentication_open_filename ()
xmlNode *	ags_xml_authentication_find_user_uuid ()
AgsXmlAuthentication *	ags_xml_authentication_new ()
#define	AGS_IS_XML_AUTHENTICATION()
#define	AGS_IS_XML_AUTHENTICATION_CLASS()
#define	AGS_XML_AUTHENTICATION()
#define	AGS_XML_AUTHENTICATION_CLASS()
#define	AGS_XML_AUTHENTICATION_GET_CLASS()
GType	ags_xml_authentication_get_type ()

Types and Values

#define	AGS_TYPE_XML_AUTHENTICATION
struct	AgsXmlAuthentication
struct	AgsXmlAuthenticationClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsXmlAuthentication
```

Implemented Interfaces

AgsXmlAuthentication implements AgsAuthentication.

Includes

```
#include <ags/server/security/ags_xml_authentication.h>
```

Description

The AgsXmlAuthentication is an object to authenticate to Advanced Gtk+ Sequencer's server.

Functions

AGS_XML_AUTHENTICATION_GET_OBJ_MUTEX()

```
#define AGS_XML_AUTHENTICATION_GET_OBJ_MUTEX(obj) (&(((AgsXmlAuthentication *) obj)-> ↩
    obj_mutex))
```

ags_xml_authentication_open_filename ()

```
void
ags_xml_authentication_open_filename (AgsXmlAuthentication *xml_authentication,
                                      gchar *filename);
```

Open *filename* .

Parameters

xml_authentication	the AgsXmlAuthentication	
filename	the filename	

Since: 3.0.0

ags_xml_authentication_find_user_uuid ()

```
xmlNode~*
ags_xml_authentication_find_user_uuid (AgsXmlAuthentication *xml_authentication,
                                       gchar *user_uuid);
```

Find ags-srv-auth xmlNode containing *user_uuid* .

Parameters

xml_authentication	the AgsXmlAuthentication	
user_uuid	the user UUID	

Returns

the matching xmlNode or NULL.

[transfer none]

Since: 3.0.0

ags_xml_authentication_new ()

```
AgsXmlAuthentication~*
ags_xml_authentication_new ();
```

Create AgsXmlAuthentication.

Returns

the new AgsXmlAuthentication instance

Since: 3.0.0

AGS_IS_XML_AUTHENTICATION()

```
#define AGS_IS_XML_AUTHENTICATION(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_XML_AUTHENTICATION))
```

AGS_IS_XML_AUTHENTICATION_CLASS()

```
#define AGS_IS_XML_AUTHENTICATION_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_XML_AUTHENTICATION))
```

AGS_XML_AUTHENTICATION()

```
#define AGS_XML_AUTHENTICATION(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_XML_AUTHENTICATION, AgsXmlAuthentication))
```

AGS_XML_AUTHENTICATION_CLASS()

```
#define AGS_XML_AUTHENTICATION_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_XML_AUTHENTICATION, AgsXmlAuthenticationClass))
```

AGS_XML_AUTHENTICATION_GET_CLASS()

```
#define AGS_XML_AUTHENTICATION_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_XML_AUTHENTICATION, AgsXmlAuthenticationClass))
```

ags_xml_authentication_get_type ()

```
GType
ags_xml_authentication_get_type ();
```

Types and Values

AGS_TYPE_XML_AUTHENTICATION

```
#define AGS_TYPE_XML_AUTHENTICATION                (ags_xml_authentication_get_type())
```

struct AgsXmlAuthentication

```
struct AgsXmlAuthentication;
```

struct AgsXmlAuthenticationClass

```
struct AgsXmlAuthenticationClass {
  GObjectClass gobject;
};
```

66.12 AgsXmlBusinessGroup

AgsXmlBusinessGroup — password store by XML file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_XML_BUSINESS_GROUP_GET_OBJ_MUTEX()
void	ags_xml_business_group_open_filename ()
AgsXmlBusinessGroup *	ags_xml_business_group_new ()
#define	AGS_IS_XML_BUSINESS_GROUP()
#define	AGS_IS_XML_BUSINESS_GROUP_CLASS()
#define	AGS_XML_BUSINESS_GROUP()
#define	AGS_XML_BUSINESS_GROUP_CLASS()
#define	AGS_XML_BUSINESS_GROUP_GET_CLASS()
GType	ags_xml_business_group_get_type ()

Types and Values

#define	AGS_TYPE_XML_BUSINESS_GROUP
struct	AgsXmlBusinessGroup
struct	AgsXmlBusinessGroupClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsXmlBusinessGroup
```

Implemented Interfaces

AgsXmlBusinessGroup implements AgsBusinessGroup.

Includes

```
#include <ags/server/security/ags_xml_business_group.h>
```

Description

The AgsXmlBusinessGroup is an object to keep passwords.

Functions

AGS_XML_BUSINESS_GROUP_GET_OBJ_MUTEX()

```
#define AGS_XML_BUSINESS_GROUP_GET_OBJ_MUTEX(obj) (&(((AgsXmlBusinessGroup *) obj)-> ↩
    obj_mutex))
```

ags_xml_business_group_open_filename ()

```
void
ags_xml_business_group_open_filename (AgsXmlBusinessGroup *xml_business_group,
                                      gchar *filename);
```

Open *filename* .

Parameters

xml_business_group	the AgsXmlBusinessGroup	
filename	the filename	

Since: 3.0.0

ags_xml_business_group_new ()

```
AgsXmlBusinessGroup~*
ags_xml_business_group_new ();
```

Create AgsXmlBusinessGroup.

Returns

the new AgsXmlBusinessGroup instance

Since: 3.0.0

AGS_IS_XML_BUSINESS_GROUP()

```
#define AGS_IS_XML_BUSINESS_GROUP(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_XML_BUSINESS_GROUP))
```

AGS_IS_XML_BUSINESS_GROUP_CLASS()

```
#define AGS_IS_XML_BUSINESS_GROUP_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_XML_BUSINESS_GROUP))
```

AGS_XML_BUSINESS_GROUP()

```
#define AGS_XML_BUSINESS_GROUP(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_XML_BUSINESS_GROUP, AgsXmlBusinessGroup))
```

AGS_XML_BUSINESS_GROUP_CLASS()

```
#define AGS_XML_BUSINESS_GROUP_CLASS(class)         (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_XML_BUSINESS_GROUP, AgsXmlBusinessGroupClass))
```

AGS_XML_BUSINESS_GROUP_GET_CLASS()

```
#define AGS_XML_BUSINESS_GROUP_GET_CLASS(obj)       (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_XML_BUSINESS_GROUP, AgsXmlBusinessGroupClass))
```

ags_xml_business_group_get_type ()

```
GType
ags_xml_business_group_get_type ();
```

Types and Values

AGS_TYPE_XML_BUSINESS_GROUP

```
#define AGS_TYPE_XML_BUSINESS_GROUP                 (ags_xml_business_group_get_type())
```

struct AgsXmlBusinessGroup

```
struct AgsXmlBusinessGroup;
```

struct AgsXmlBusinessGroupClass

```
struct AgsXmlBusinessGroupClass {
  GObjectClass gobject;
};
```

66.13 AgsXmlCertificate

AgsXmlCertificate — certificate by XML file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_XML_CERTIFICATE_GET_OBJ_MUTEX()
void	ags_xml_certificate_open_filename ()
AgsXmlCertificate *	ags_xml_certificate_new ()
#define	AGS_IS_XML_CERTIFICATE()
#define	AGS_IS_XML_CERTIFICATE_CLASS()
#define	AGS_XML_CERTIFICATE()
#define	AGS_XML_CERTIFICATE_CLASS()
#define	AGS_XML_CERTIFICATE_GET_CLASS()
GType	ags_xml_certificate_get_type ()

Types and Values

#define	AGS_TYPE_XML_CERTIFICATE
struct	AgsXmlCertificate
struct	AgsXmlCertificateClass

Object Hierarchy

```
GObject
&#x2570;&#x2500;&#x2500; AgsXmlCertificate
```

Implemented Interfaces

AgsXmlCertificate implements AgsCertificate.

Includes

```
#include <ags/server/security/ags_xml_certificate.h>
```

Description

The AgsXmlCertificate is an object to verify certificates.

Functions

AGS_XML_CERTIFICATE_GET_OBJ_MUTEX()

```
#define AGS_XML_CERTIFICATE_GET_OBJ_MUTEX(obj) (&(((AgsXmlCertificate *) obj)->obj_mutex))
```

ags_xml_certificate_open_filename ()

```
void
ags_xml_certificate_open_filename (AgsXmlCertificate *xml_certificate,
                                   gchar *filename);
```

Open *filename*.

Parameters

xml_certificate	the AgsXmlCertificate	
filename	the filename	

Since: 3.0.0

ags_xml_certificate_new ()

```
AgsXmlCertificate~*
ags_xml_certificate_new ();
```

Create AgsXmlCertificate.

Returns

the new AgsXmlCertificate instance

Since: 3.0.0

AGS_IS_XML_CERTIFICATE()

```
#define AGS_IS_XML_CERTIFICATE(obj)             (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_XML_CERTIFICATE))
```

AGS_IS_XML_CERTIFICATE_CLASS()

```
#define AGS_IS_XML_CERTIFICATE_CLASS(class)     (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_XML_CERTIFICATE))
```

AGS_XML_CERTIFICATE()

```
#define AGS_XML_CERTIFICATE(obj)                (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_XML_CERTIFICATE, AgsXmlCertificate))
```

AGS_XML_CERTIFICATE_CLASS()

```
#define AGS_XML_CERTIFICATE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_XML_CERTIFICATE, AgsXmlCertificateClass))
```

AGS_XML_CERTIFICATE_GET_CLASS()

```
#define AGS_XML_CERTIFICATE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_XML_CERTIFICATE, AgsXmlCertificateClass))
```

ags_xml_certificate_get_type ()

```
GType
ags_xml_certificate_get_type ();
```

Types and Values

AGS_TYPE_XML_CERTIFICATE

```
#define AGS_TYPE_XML_CERTIFICATE                (ags_xml_certificate_get_type())
```

struct AgsXmlCertificate

```
struct AgsXmlCertificate;
```

struct AgsXmlCertificateClass

```
struct AgsXmlCertificateClass {
  GObjectClass gobject;
};
```

66.14 AgsXmlPasswordStore

AgsXmlPasswordStore — password store by XML file

Stability Level

Stable, unless otherwise indicated

Functions

#define	AGS_XML_PASSWORD_STORE_GET_OBJ_MUTEX()
void	ags_xml_password_store_open_filename ()
xmlNode *	ags_xml_password_store_find_login ()
AgsXmlPasswordStore *	ags_xml_password_store_new ()
#define	AGS_IS_XML_PASSWORD_STORE()
#define	AGS_IS_XML_PASSWORD_STORE_CLASS()
#define	AGS_XML_PASSWORD_STORE()
#define	AGS_XML_PASSWORD_STORE_CLASS()
#define	AGS_XML_PASSWORD_STORE_GET_CLASS()
GType	ags_xml_password_store_get_type ()

Types and Values

#define	AGS_TYPE_XML_PASSWORD_STORE
struct	AgsXmlPasswordStore
struct	AgsXmlPasswordStoreClass

Object Hierarchy

```
    GObject
    &#x2570;&#x2500;&#x2500; AgsXmlPasswordStore
```

Implemented Interfaces

AgsXmlPasswordStore implements AgsPasswordStore.

Includes

```
#include <ags/server/security/ags_xml_password_store.h>
```

Description

The AgsXmlPasswordStore is an object to keep passwords.

Functions

AGS_XML_PASSWORD_STORE_GET_OBJ_MUTEX()

```
#define AGS_XML_PASSWORD_STORE_GET_OBJ_MUTEX(obj) (&(((AgsXmlPasswordStore *) obj)-> ↩
    obj_mutex))
```

ags_xml_password_store_open_filename ()

```
void
ags_xml_password_store_open_filename (AgsXmlPasswordStore *xml_password_store,
                                      gchar *filename);
```

Open *filename* .

Parameters

xml_password_store	the AgsXmlPasswordStore	
filename	the filename	

Since: 3.0.0

ags_xml_password_store_find_login ()

```
xmlNode~*
ags_xml_password_store_find_login (AgsXmlPasswordStore *xml_password_store,
                                   gchar *login);
```

Find ags-srv-user xmlNode containing *login* .

Parameters

xml_password_store	the AgsXmlPasswordStore	
login	the login	

Returns

the matching xmlNode or NULL.

[transfer none]

Since: 3.0.0

ags_xml_password_store_new ()

```
AgsXmlPasswordStore~*
ags_xml_password_store_new ();
```

Create AgsXmlPasswordStore.

Returns

the new AgsXmlPasswordStore instance

Since: 3.0.0

AGS_IS_XML_PASSWORD_STORE()

```
#define AGS_IS_XML_PASSWORD_STORE(obj)              (G_TYPE_CHECK_INSTANCE_TYPE ((obj),  ↩
    AGS_TYPE_XML_PASSWORD_STORE))
```

AGS_IS_XML_PASSWORD_STORE_CLASS()

```
#define AGS_IS_XML_PASSWORD_STORE_CLASS(class)      (G_TYPE_CHECK_CLASS_TYPE ((class),  ↩
    AGS_TYPE_XML_PASSWORD_STORE))
```

AGS_XML_PASSWORD_STORE()

```
#define AGS_XML_PASSWORD_STORE(obj)                 (G_TYPE_CHECK_INSTANCE_CAST((obj),  ↩
    AGS_TYPE_XML_PASSWORD_STORE, AgsXmlPasswordStore))
```

AGS_XML_PASSWORD_STORE_CLASS()

```
#define AGS_XML_PASSWORD_STORE_CLASS(class)        (G_TYPE_CHECK_CLASS_CAST(class,  ↩
    AGS_TYPE_XML_PASSWORD_STORE, AgsXmlPasswordStoreClass))
```

AGS_XML_PASSWORD_STORE_GET_CLASS()

```
#define AGS_XML_PASSWORD_STORE_GET_CLASS(obj)      (G_TYPE_INSTANCE_GET_CLASS(obj,  ↩
    AGS_TYPE_XML_PASSWORD_STORE, AgsXmlPasswordStoreClass))
```

ags_xml_password_store_get_type ()

```
GType
ags_xml_password_store_get_type ();
```

Types and Values

AGS_TYPE_XML_PASSWORD_STORE

```
#define AGS_TYPE_XML_PASSWORD_STORE                (ags_xml_password_store_get_type())
```

struct AgsXmlPasswordStore

```
struct AgsXmlPasswordStore;
```

struct AgsXmlPasswordStoreClass

```
struct AgsXmlPasswordStoreClass {
  GObjectClass gobject;
};
```

Part VII

Annotation Glossary

A

array

Parameter points to an array of items.

E

element-type

Generics and defining elements of containers and arrays.

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 callee-allocates

Out parameter, where callee 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 VIII

Tree Index

```
GObject
&#x251c;&#x2500;&#x2500; AgsApplicationContext
&#x2502;   &#x251c;&#x2500;&#x2500; AgsServerApplicationContext
&#x2502;   &#x2570;&#x2500;&#x2500; AgsThreadApplicationContext
&#x251c;&#x2500;&#x2500; AgsSecurityContext
&#x2502;   &#x2570;&#x2500;&#x2500; AgsAuthSecurityContext
&#x251c;&#x2500;&#x2500; AgsAuthenticationManager
&#x251c;&#x2500;&#x2500; AgsBusinessGroupManager
&#x251c;&#x2500;&#x2500; AgsCertificateManager
&#x251c;&#x2500;&#x2500; AgsConfig
&#x251c;&#x2500;&#x2500; AgsController
&#x2502;   &#x2570;&#x2500;&#x2500; AgsFrontController
&#x251c;&#x2500;&#x2500; AgsConversion
&#x251c;&#x2500;&#x2500; AgsThread
&#x2502;   &#x251c;&#x2500;&#x2500; AgsWorkerThread
&#x2502;   &#x2502;   &#x2570;&#x2500;&#x2500; AgsDestroyWorker
&#x2502;   &#x251c;&#x2500;&#x2500; AgsGenericMainLoop
&#x2502;   &#x2570;&#x2500;&#x2500; AgsReturnableThread
&#x251c;&#x2500;&#x2500; AgsFile
&#x251c;&#x2500;&#x2500; AgsFileIdRef
&#x251c;&#x2500;&#x2500; AgsFileLaunch
&#x251c;&#x2500;&#x2500; AgsFileLink
&#x251c;&#x2500;&#x2500; AgsFileLookup
&#x251c;&#x2500;&#x2500; AgsFunction
&#x251c;&#x2500;&#x2500; AgsLog
&#x251c;&#x2500;&#x2500; AgsMessageDelivery
&#x251c;&#x2500;&#x2500; AgsMessageEnvelope
&#x251c;&#x2500;&#x2500; AgsMessageQueue
&#x251c;&#x2500;&#x2500; AgsPasswordStoreManager
&#x251c;&#x2500;&#x2500; AgsPriority
&#x251c;&#x2500;&#x2500; AgsRegistry
&#x251c;&#x2500;&#x2500; AgsServer
&#x251c;&#x2500;&#x2500; AgsServerStatus
&#x251c;&#x2500;&#x2500; AgsSolverMatrix
&#x251c;&#x2500;&#x2500; AgsSolverPolynomial
&#x251c;&#x2500;&#x2500; AgsSolverVector
&#x251c;&#x2500;&#x2500; AgsTaskCompletion
&#x251c;&#x2500;&#x2500; AgsTask
&#x251c;&#x2500;&#x2500; AgsTaskLauncher
&#x251c;&#x2500;&#x2500; AgsThreadPool
&#x251c;&#x2500;&#x2500; AgsTimestamp
&#x251c;&#x2500;&#x2500; AgsTurtle
&#x251c;&#x2500;&#x2500; AgsTurtleManager
&#x251c;&#x2500;&#x2500; AgsXmlAuthentication
&#x251c;&#x2500;&#x2500; AgsXmlBusinessGroup
&#x251c;&#x2500;&#x2500; AgsXmlCertificate
&#x2570;&#x2500;&#x2500; AgsXmlPasswordStore
GInterface
&#x251c;&#x2500;&#x2500; AgsApplicable
&#x251c;&#x2500;&#x2500; AgsConnectable
&#x251c;&#x2500;&#x2500; AgsAuthentication
&#x251c;&#x2500;&#x2500; AgsBusinessGroup
&#x251c;&#x2500;&#x2500; AgsCertificate
&#x251c;&#x2500;&#x2500; AgsConcurrencyProvider
&#x251c;&#x2500;&#x2500; AgsCountable
&#x251c;&#x2500;&#x2500; AgsCursor
&#x251c;&#x2500;&#x2500; AgsPlugin
&#x251c;&#x2500;&#x2500; AgsMainLoop
&#x251c;&#x2500;&#x2500; AgsMutable
&#x251c;&#x2500;&#x2500; AgsPasswordStore
&#x251c;&#x2500;&#x2500; AgsPluginController
```

```
    &#x251c;&#x2500;&#x2500; AgsPortlet
    &#x251c;&#x2500;&#x2500; AgsSeekable
    &#x251c;&#x2500;&#x2500; AgsSequencer
    &#x251c;&#x2500;&#x2500; AgsServiceProvider
    &#x251c;&#x2500;&#x2500; AgsSoundServer
    &#x251c;&#x2500;&#x2500; AgsSoundcard
    &#x2570;&#x2500;&#x2500; AgsTactable
    GFlags
    &#x251c;&#x2500;&#x2500; AgsApplicationContextFlags
    &#x251c;&#x2500;&#x2500; AgsFileFlags
    &#x251c;&#x2500;&#x2500; AgsReturnableThreadFlags
    &#x251c;&#x2500;&#x2500; AgsSecurityContextMode
    &#x251c;&#x2500;&#x2500; AgsServerFlags
    &#x251c;&#x2500;&#x2500; AgsSoundcardCapability
    &#x251c;&#x2500;&#x2500; AgsTaskCompletionFlags
    &#x251c;&#x2500;&#x2500; AgsThreadApplicationContextFlags
    &#x251c;&#x2500;&#x2500; AgsThreadFlags
    &#x251c;&#x2500;&#x2500; AgsThreadPoolFlags
    &#x251c;&#x2500;&#x2500; AgsThreadStatusFlags
    &#x251c;&#x2500;&#x2500; AgsThreadSyncTicFlags
    &#x251c;&#x2500;&#x2500; AgsTimestampFlags
    &#x251c;&#x2500;&#x2500; AgsTurtleFlags
    &#x2570;&#x2500;&#x2500; AgsWorkerThreadStatusFlags
    GBoxed
    &#x251c;&#x2500;&#x2500; AgsBufferUtil
    &#x251c;&#x2500;&#x2500; AgsComplex
    &#x251c;&#x2500;&#x2500; AgsDestroyUtil
    &#x251c;&#x2500;&#x2500; AgsListUtil
    &#x251c;&#x2500;&#x2500; AgsMathUtil
    &#x251c;&#x2500;&#x2500; AgsStringUtil
    &#x2570;&#x2500;&#x2500; AgsUUID
    GEnum
    &#x251c;&#x2500;&#x2500; AgsByteOrder
    &#x251c;&#x2500;&#x2500; AgsSeekType
    &#x2570;&#x2500;&#x2500; AgsSoundcardFormat
```

Chapter 67

Index

A
AGS_APPLICABLE, 132
AgsApplicable, 132
ags_applicable_apply, 131
AGS_APPLICABLE_GET_INTERFACE, 132
ags_applicable_get_type, 132
AGS_APPLICABLE_INTERFACE, 132
ags_applicable_reset, 131
ags_applicable_set_update, 131
AgsApplicableInterface, 133
AGS_APPLICATION_CONTEXT, 139
AGS_APPLICATION_CONTEXT_CLASS, 139
ags_application_context_flags_get_type, 139
AGS_APPLICATION_CONTEXT_GET_CLASS, 139
ags_application_context_get_instance, 138
AGS_APPLICATION_CONTEXT_GET_OBJ_MUTEX, 136
ags_application_context_get_type, 139
ags_application_context_load_config, 137
ags_application_context_new, 138
ags_application_context_prepare, 137
ags_application_context_quit, 138
ags_application_context_register_types, 138
ags_application_context_set_flags, 136
ags_application_context_setup, 137
ags_application_context_test_flags, 136
ags_application_context_unset_flags, 136
AgsApplicationContext, 140
AgsApplicationContext::load-config, 142
AgsApplicationContext::prepare, 142
AgsApplicationContext::quit, 143
AgsApplicationContext::register-types, 143
AgsApplicationContext::setup, 143
AgsApplicationContext:config, 141
AgsApplicationContext:file, 141
AgsApplicationContext:main-loop, 141
AgsApplicationContext:task-launcher, 142
AgsApplicationContextClass, 141
AgsApplicationContextFlags, 140
AGS_AUTH_SECURITY_CONTEXT, 536
AGS_AUTH_SECURITY_CONTEXT_CLASS, 536
AGS_AUTH_SECURITY_CONTEXT_GET_CLASS, 536
ags_auth_security_context_get_instance, 536
AGS_AUTH_SECURITY_CONTEXT_GET_OBJ_MUTEX, 536
ags_auth_security_context_get_type, 537
ags_auth_security_context_new, 536
AGS_AUTHENTICATION, 495
AgsAuthentication, 496
ags_authentication_generate_token, 493
ags_authentication_get_authentication_module, 492
ags_authentication_get_digest, 494
AGS_AUTHENTICATION_GET_INTERFACE, 495
ags_authentication_get_type, 495
AGS_AUTHENTICATION_INTERFACE, 495
ags_authentication_is_session_active, 494
ags_authentication_login, 492
ags_authentication_logout, 493
AGS_AUTHENTICATION_MANAGER, 504
ags_authentication_manager_add_authentication, 499
AGS_AUTHENTICATION_MANAGER_CLASS, 504
AGS_AUTHENTICATION_MANAGER_DEFAULT_SESSION_T 504
ags_authentication_manager_get_authentication, 499
AGS_AUTHENTICATION_MANAGER_GET_CLASS, 504
ags_authentication_manager_get_digest, 502
ags_authentication_manager_get_instance, 503
AGS_AUTHENTICATION_MANAGER_GET_OBJ_MUTEX, 498
ags_authentication_manager_get_session_timeout, 500
ags_authentication_manager_get_type, 504
ags_authentication_manager_insert_login, 501
ags_authentication_manager_is_session_active, 503
ags_authentication_manager_login, 501
ags_authentication_manager_logout, 502
ags_authentication_manager_lookup_login, 500
ags_authentication_manager_new, 504
ags_authentication_manager_remove_authentication, 499
ags_authentication_manager_remove_login, 501
AgsAuthenticationInterface, 496
AgsAuthenticationManager, 505
AgsAuthenticationManagerClass, 505
AgsAuthSecurityContext, 537
AgsAuthSecurityContextClass, 537

B

ags_buffer_util_char_buffer_read_complex, 12
ags_buffer_util_char_buffer_read_double, 12
ags_buffer_util_char_buffer_read_float, 11
ags_buffer_util_char_buffer_read_s16, 10
ags_buffer_util_char_buffer_read_s24, 10
ags_buffer_util_char_buffer_read_s32, 11
ags_buffer_util_char_buffer_read_s64, 11
ags_buffer_util_char_buffer_read_s8, 10
ags_buffer_util_char_buffer_swap_bytes, 15
ags_buffer_util_char_buffer_to_complex, 9
ags_buffer_util_char_buffer_to_double, 9
ags_buffer_util_char_buffer_to_float, 8
ags_buffer_util_char_buffer_to_s16, 7
ags_buffer_util_char_buffer_to_s24, 7
ags_buffer_util_char_buffer_to_s32, 8
ags_buffer_util_char_buffer_to_s64, 8
ags_buffer_util_char_buffer_to_s8, 6
ags_buffer_util_char_buffer_write_complex, 15
ags_buffer_util_char_buffer_write_double, 15
ags_buffer_util_char_buffer_write_float, 14
ags_buffer_util_char_buffer_write_s16, 13
ags_buffer_util_char_buffer_write_s24, 13
ags_buffer_util_char_buffer_write_s32, 14
ags_buffer_util_char_buffer_write_s64, 14
ags_buffer_util_char_buffer_write_s8, 13
ags_buffer_util_complex_to_char_buffer, 6
ags_buffer_util_double_to_char_buffer, 6
ags_buffer_util_float_to_char_buffer, 5
ags_buffer_util_get_type, 16
ags_buffer_util_s16_to_char_buffer, 4
ags_buffer_util_s24_to_char_buffer, 4
ags_buffer_util_s32_to_char_buffer, 4
ags_buffer_util_s64_to_char_buffer, 5
ags_buffer_util_s8_to_char_buffer, 3
AgsBufferUtil, 16
AGS_BUILD_ID, 140
AGS_BUSINESS_GROUP, 509
ags_business_group_get_group_name, 507
ags_business_group_get_group_uuid, 506
AGS_BUSINESS_GROUP_GET_INTERFACE, 509
ags_business_group_get_type, 509
ags_business_group_get_user, 508
AGS_BUSINESS_GROUP_INTERFACE, 509
AGS_BUSINESS_GROUP_MANAGER, 513
ags_business_group_manager_add_business_group, 512
AGS_BUSINESS_GROUP_MANAGER_CLASS, 513
ags_business_group_manager_get_business_group, 511
AGS_BUSINESS_GROUP_MANAGER_GET_CLASS, 513
ags_business_group_manager_get_instance, 512
AGS_BUSINESS_GROUP_MANAGER_GET_OBJ_MUTEX, 511
ags_business_group_manager_get_type, 513
ags_business_group_manager_new, 513
ags_business_group_manager_remove_business_group, 512
ags_business_group_set_group_name, 507
ags_business_group_set_user, 508
AgsBusinessGroup, 509
AgsBusinessGroupInterface, 510
AgsBusinessGroupManager, 514
AgsBusinessGroupManagerClass, 514
ags_byte_order_get_type, 27
AgsByteOrder, 28

C

ags_cclosure_marshal_BOOLEAN__OBJECT_OBJECT, 203
ags_cclosure_marshal_BOOLEAN__POINTER_UINT, 203
ags_cclosure_marshal_BOOLEAN__UINT_POINTER, 203
ags_cclosure_marshal_BOOLEAN__VOID, 203
ags_cclosure_marshal_DOUBLE__DOUBLE_BOOLEAN, 204
ags_cclosure_marshal_INT64__OBJECT, 202
ags_cclosure_marshal_INT__VOID, 202
ags_cclosure_marshal_OBJECT__OBJECT, 210
ags_cclosure_marshal_OBJECT__OBJECT_OBJECT, 210
ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER 210
ags_cclosure_marshal_OBJECT__OBJECT_POINTER_POINTER 210
ags_cclosure_marshal_OBJECT__STRING_STRING_STRING, 210
ags_cclosure_marshal_OBJECT__UINT_POINTER, 209
ags_cclosure_marshal_OBJECT__VOID, 209
ags_cclosure_marshal_POINTER__INT, 204
ags_cclosure_marshal_POINTER__INT_UINT, 205
ags_cclosure_marshal_POINTER__OBJECT, 207
ags_cclosure_marshal_POINTER__OBJECT_OBJECT, 207
ags_cclosure_marshal_POINTER__OBJECT_POINTER_INT64, 207
ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT 209
ags_cclosure_marshal_POINTER__OBJECT_POINTER_OBJECT 208
ags_cclosure_marshal_POINTER__OBJECT_POINTER_POINTE 209
ags_cclosure_marshal_POINTER__OBJECT_POINTER_UINT, 207
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_ 208
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_ 208
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_ 208
ags_cclosure_marshal_POINTER__OBJECT_STRING_STRING_ 208
ags_cclosure_marshal_POINTER__POINTER, 205
ags_cclosure_marshal_POINTER__POINTER_POINTER_POINT 205
ags_cclosure_marshal_POINTER__POINTER_POINTER_UINT, 206
ags_cclosure_marshal_POINTER__POINTER_STRING_STRING 209
ags_cclosure_marshal_POINTER__POINTER_UINT, 206
ags_cclosure_marshal_POINTER__POINTER_UINT_POINTER_ 207

ags_cclosure_marshal_POINTER__STRING_STRING, 206
ags_cclosure_marshal_POINTER__STRING_STRING_STRING, 206
ags_cclosure_marshal_POINTER__STRING_STRING_STRING_STRING, 206
ags_cclosure_marshal_POINTER__UINT, 204
ags_cclosure_marshal_POINTER__UINT_POINTER, 205
ags_cclosure_marshal_POINTER__UINT_UINT, 205
ags_cclosure_marshal_POINTER__VOID, 202
ags_cclosure_marshal_STRING__POINTER_STRING_STRING_STRING_UINT_POINTER, 204
ags_cclosure_marshal_STRING__STRING_STRING, 204
ags_cclosure_marshal_UINT__INT_UINT_POINTER, 203
ags_cclosure_marshal_UINT__VOID, 202
ags_cclosure_marshal_VOID__BOOLEAN_POINTER, 193
ags_cclosure_marshal_VOID__DOUBLE_DOUBLE, 197
ags_cclosure_marshal_VOID__INT64_UINT, 197
ags_cclosure_marshal_VOID__INT_POINTER, 194
ags_cclosure_marshal_VOID__INT_UINT, 193
ags_cclosure_marshal_VOID__INT_UINT_POINTER, 194
ags_cclosure_marshal_VOID__OBJECT_OBJECT, 201
ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_INT_BOOLEAN, 201
ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_OBJECT, 202
ags_cclosure_marshal_VOID__OBJECT_OBJECT_OBJECT_OBJECT_UINT, 201
ags_cclosure_marshal_VOID__OBJECT_UINT, 201
ags_cclosure_marshal_VOID__OBJECT_UINT_UINT_UINT, 201
ags_cclosure_marshal_VOID__POINTER_INT, 199
ags_cclosure_marshal_VOID__POINTER_INT_INT_BOOLEAN, 199
ags_cclosure_marshal_VOID__POINTER_INT_POINTER, 199
ags_cclosure_marshal_VOID__POINTER_OBJECT_OBJECT_STRING_STRING_STRING_UINT_UINT_UINT_UINT_INT_UINT, 200
ags_cclosure_marshal_VOID__POINTER_POINTER, 199
ags_cclosure_marshal_VOID__POINTER_POINTER_POINTER, 200
ags_cclosure_marshal_VOID__POINTER_POINTER_UINT, 200
ags_cclosure_marshal_VOID__POINTER_STRING_STRING_STRING_INT, 200
ags_cclosure_marshal_VOID__POINTER_UINT, 197
ags_cclosure_marshal_VOID__POINTER_UINT_BOOLEAN, 197
ags_cclosure_marshal_VOID__POINTER_UINT_POINTER, 200
ags_cclosure_marshal_VOID__POINTER_UINT_UINT, 197
ags_cclosure_marshal_VOID__STRING_POINTER, 198
ags_cclosure_marshal_VOID__STRING_STRING_STRING, 199
ags_cclosure_marshal_VOID__STRING_UINT, 198
ags_cclosure_marshal_VOID__STRING_UINT_DOUBLE, 198
ags_cclosure_marshal_VOID__STRING_UINT_POINTER, 198
ags_cclosure_marshal_VOID__STRING_UINT_STRING_DOUB 198
ags_cclosure_marshal_VOID__UINT64, 193
ags_cclosure_marshal_VOID__UINT_BOOLEAN, 194
ags_cclosure_marshal_VOID__UINT_DOUBLE, 196
ags_cclosure_marshal_VOID__UINT_DOUBLE_UINT, 196
ags_cclosure_marshal_VOID__UINT_INT, 196
ags_cclosure_marshal_VOID__UINT_POINTER, 196
ags_cclosure_marshal_VOID__UINT_POINTER_UINT, 196
ags_cclosure_marshal_VOID__UINT_STRING_UINT, 196
ags_cclosure_marshal_VOID__UINT_UINT, 194
ags_cclosure_marshal_VOID__UINT_UINT_BOOLEAN, 194
ags_cclosure_marshal_VOID__UINT_UINT_UINT, 195
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT, 195
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT 195
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT 195
ags_cclosure_marshal_VOID__UINT_UINT_UINT_UINT_UINT 195
AGS_CERTIFICATE, 520
AGS_CERTIFICATE, 520
ags_certificate_get_cert_uid, 515
ags_certificate_get_domain, 516
AGS_CERTIFICATE_GET_INTERFACE, 520
ags_certificate_get_key_type, 517
ags_certificate_get_private_key_file, 519
ags_certificate_get_public_key_file, 518
ags_certificate_get_type, 520
AGS_CERTIFICATE_INTERFACE, 520
AGS_CERTIFICATE_MANAGER, 525
ags_certificate_manager_add_certificate, 523
AGS_CERTIFICATE_MANAGER_CLASS, 525
ags_certificate_manager_get_certificate, 523
AGS_CERTIFICATE_MANAGER_GET_CLASS, 525
ags_certificate_manager_get_instance, 524
ags_certificate_manager_get_key_file, 524
AGS_CERTIFICATE_MANAGER_GET_OBJ_MUTEX, 523
ags_certificate_manager_get_type, 525
ags_certificate_manager_new, 525
ags_certificate_manager_remove_certificate, 523
ags_certificate_set_domain, 516
ags_certificate_set_key_type, 517
ags_certificate_set_private_key_file, 519
ags_certificate_set_public_key_file, 518
AgsCertificateInterface, 521
AgsCertificateManager, 526
AgsCertificateManagerClass, 526
AGS_COMPLEX, 19
AgsComplex, 20
ags_complex_alloc, 18
ags_complex_copy, 18
ags_complex_free, 18
ags_complex_get_term, 19
ags_complex_get_type, 19

AGS_COMPLEX_M_E, 19
AGS_COMPLEX_M_PI, 19
ags_complex_set, 19
ags_complex_set_term, 19
AGS_CONCURRENCY_PROVIDER, 339
AGS_CONCURRENCY_PROVIDER_GET_INTERFACE, 339
ags_concurrency_provider_get_main_loop, 336
ags_concurrency_provider_get_task_launcher, 337
ags_concurrency_provider_get_thread_pool, 337
ags_concurrency_provider_get_type, 339
ags_concurrency_provider_get_worker, 338
AGS_CONCURRENCY_PROVIDER_INTERFACE, 339
ags_concurrency_provider_set_main_loop, 336
ags_concurrency_provider_set_task_launcher, 337
ags_concurrency_provider_set_thread_pool, 338
ags_concurrency_provider_set_worker, 339
AgsConcurrencyProvider, 340
AgsConcurrencyProviderInterface, 340
AGS_CONFIG, 150
AgsConfig, 152
AgsConfig::get-value, 152
AgsConfig::load-defaults, 152
AgsConfig::set-value, 153
AGS_CONFIG_CLASS, 150
ags_config_clear, 149
AGS_CONFIG_DEFAULT_BUILD_ID, 150
AGS_CONFIG_DEFAULT_VERSION, 150
AGS_CONFIG_GENERIC, 150
AGS_CONFIG_GET_CLASS, 150
ags_config_get_instance, 149
AGS_CONFIG_GET_OBJ_MUTEX, 146
ags_config_get_type, 150
ags_config_get_value, 148
ags_config_load_defaults, 146
ags_config_load_from_data, 147
ags_config_load_from_file, 147
ags_config_new, 149
AGS_CONFIG_OSC_SERVER, 151
AGS_CONFIG_OSC_SERVER_0, 151
AGS_CONFIG_RECALL, 151
ags_config_save, 149
AGS_CONFIG_SEQUENCER, 151
AGS_CONFIG_SEQUENCER_0, 151
AGS_CONFIG_SERVER, 151
ags_config_set_value, 147
AGS_CONFIG_SOUNDCARD, 151
AGS_CONFIG_SOUNDCARD_0, 151
AGS_CONFIG_THREAD, 151
ags_config_to_data, 148
AgsConfigClass, 152
AGS_CONNECTABLE, 167
AgsConnectable, 168
ags_connectable_add_to_registry, 164
ags_connectable_connect, 166
ags_connectable_connect_connection, 167
ags_connectable_disconnect, 166
ags_connectable_disconnect_connection, 167
AGS_CONNECTABLE_GET_INTERFACE, 167
ags_connectable_get_type, 168
ags_connectable_get_uuid, 163
ags_connectable_has_resource, 163
AGS_CONNECTABLE_INTERFACE, 167
ags_connectable_is_connected, 166
ags_connectable_is_ready, 164
ags_connectable_list_resource, 165
ags_connectable_remove_from_registry, 164
ags_connectable_xml_compose, 165
ags_connectable_xml_parse, 165
AgsConnectableFlags, 168
AgsConnectableInterface, 168
AGS_CONTROLLER, 482
AgsController, 484
AgsController:context-path, 484
AgsController:server, 484
ags_controller_add_resource, 480
AGS_CONTROLLER_BASE_PATH, 483
AGS_CONTROLLER_CLASS, 482
AGS_CONTROLLER_GET_CLASS, 482
AGS_CONTROLLER_GET_OBJ_MUTEX, 479
ags_controller_get_type, 483
ags_controller_lookup_resource, 481
ags_controller_new, 482
ags_controller_query_security_context, 481
ags_controller_remove_resource, 481
ags_controller_resource_alloc, 479
ags_controller_resource_free, 480
ags_controller_resource_ref, 480
ags_controller_resource_unref, 480
AgsControllerClass, 484
AgsControllerResource, 483
AGS_CONVERSION, 23
AgsConversion, 24
AgsConversion::convert, 24
AgsConversion:description, 24
AgsConversion:name, 24
AGS_CONVERSION_CLASS, 23
ags_conversion_convert, 22
AGS_CONVERSION_GET_CLASS, 23
AGS_CONVERSION_GET_OBJ_MUTEX, 22
ags_conversion_get_type, 23
ags_conversion_new, 23
AgsConversionClass, 24
AGS_COUNTABLE, 172
AgsCountable, 173
AGS_COUNTABLE_GET_INTERFACE, 172
ags_countable_get_midi_counter, 172
ags_countable_get_notation_counter, 171
ags_countable_get_sequencer_counter, 171
ags_countable_get_type, 173
ags_countable_get_wave_counter, 171
AGS_COUNTABLE_INTERFACE, 172
AgsCountableInterface, 173
AGS_CURSOR, 179

AgsCursor, 180
ags_cursor_get_bpm, 176
ags_cursor_get_current, 179
ags_cursor_get_current_copy, 178
ags_cursor_get_default_offset, 175
ags_cursor_get_delay, 176
ags_cursor_get_delay_counter, 177
ags_cursor_get_duration, 177
AGS_CURSOR_GET_INTERFACE, 179
ags_cursor_get_next, 178
ags_cursor_get_offset, 177
ags_cursor_get_prev, 178
ags_cursor_get_rate, 176
ags_cursor_get_tact, 175
ags_cursor_get_type, 180
AGS_CURSOR_INTERFACE, 179
AgsCursorInterface, 180

D
AGS_DECLARE_INTERFACE, 183
AGS_DEFAULT_CONFIG, 140
AGS_DEFAULT_DIRECTORY, 140
AGS_DESTROY_ENTRY, 342
ags_destroy_entry_alloc, 342
ags_destroy_util_dispose_and_unref, 122
ags_destroy_util_get_type, 122
AGS_DESTROY_WORKER, 343
ags_destroy_worker_add, 343
AGS_DESTROY_WORKER_CLASS, 344
AGS_DESTROY_WORKER_GET_CLASS, 344
ags_destroy_worker_get_instance, 343
ags_destroy_worker_get_type, 344
ags_destroy_worker_new, 343
AgsDestroyEntry, 344
AgsDestroyFunc, 342
AgsDestroyUtil, 122
AgsDestroyWorker, 345
AgsDestroyWorkerClass, 345

E
ags_endian_host_is_be, 27
ags_endian_host_is_le, 27
ags_endian_swap_float, 27

F
AGS_FILE, 305
AgsFile, 308
AgsFile::open, 310
AgsFile::open-from-data, 310
AgsFile::read, 311
AgsFile::read-resolve, 311
AgsFile::read-start, 311
AgsFile::rw-open, 312
AgsFile::write, 312
AgsFile::write-concurrent, 312
AgsFile::write-resolve, 312
AgsFile:audio-encoding, 309
AgsFile:audio-format, 309
AgsFile:encoding, 309
AgsFile:filename, 309
AgsFile:xml-doc, 310
ags_file_add_id_ref, 295
ags_file_add_launch, 297
ags_file_add_lookup, 297
AGS_FILE_CHARSET_CONVERTER_MAX_CONTENT_LENG
307
AGS_FILE_CHARSET_CONVERTER_MAX_STRING_LENGT
307
AGS_FILE_CHECKSUM_LENGTH, 306
AGS_FILE_CLASS, 305
ags_file_close, 303
AGS_FILE_DEFAULT_APP_ENCODING, 306
AGS_FILE_DEFAULT_AUDIO_ENCODING, 307
AGS_FILE_DEFAULT_AUDIO_FORMAT, 307
AGS_FILE_DEFAULT_DTD, 306
AGS_FILE_DEFAULT_ENCODING, 306
AGS_FILE_ERROR, 308
ags_file_find_id_ref_by_node, 296
ags_file_find_id_ref_by_reference, 297
ags_file_find_id_ref_by_xpath, 296
ags_file_flags_get_type, 306
ags_file_get_audio_encoding, 294
ags_file_get_audio_format, 293
AGS_FILE_GET_CLASS, 306
ags_file_get_encoding, 293
ags_file_get_filename, 292
AGS_FILE_GET_OBJ_MUTEX, 291
ags_file_get_type, 306
ags_file_get_xml_doc, 295
AGS_FILE_ID_REF, 316
AGS_FILE_ID_REF_CLASS, 316
AGS_FILE_ID_REF_GET_CLASS, 316
AGS_FILE_ID_REF_GET_OBJ_MUTEX, 315
ags_file_id_ref_get_type, 316
ags_file_id_ref_new, 316
AGS_FILE_ID_REF_RESOLVE_DATA, 317
ags_file_id_ref_resolved, 315
AGS_FILE_ID_REF_SERIALIZE_DATA, 317
AGS_FILE_LAUNCH, 331
AGS_FILE_LAUNCH_CLASS, 331
AGS_FILE_LAUNCH_GET_CLASS, 331
AGS_FILE_LAUNCH_GET_OBJ_MUTEX, 330
ags_file_launch_get_type, 331
ags_file_launch_new, 330
ags_file_launch_start, 330
AGS_FILE_LINK, 322
AGS_FILE_LINK_CLASS, 322
AGS_FILE_LINK_DEFAULT_BUILD_ID, 322
AGS_FILE_LINK_DEFAULT_VERSION, 322
AGS_FILE_LINK_GET_CLASS, 322
ags_file_link_get_data, 321
ags_file_link_get_filename, 320
AGS_FILE_LINK_GET_OBJ_MUTEX, 320
ags_file_link_get_type, 322

ags_file_link_new, 321
ags_file_link_set_data, 321
ags_file_link_set_filename, 320
AGS_FILE_LOOKUP, 326
AGS_FILE_LOOKUP_CLASS, 327
ags_file_lookup_find_by_node, 325
ags_file_lookup_find_by_reference, 326
AGS_FILE_LOOKUP_GET_CLASS, 327
AGS_FILE_LOOKUP_GET_OBJ_MUTEX, 325
ags_file_lookup_get_type, 327
ags_file_lookup_new, 326
ags_file_lookup_resolve, 326
ags_file_new, 305
ags_file_open, 302
ags_file_open_filename, 303
ags_file_open_from_data, 302
ags_file_read, 304
ags_file_read_application_context, 305
ags_file_read_config, 305
ags_file_read_resolve, 304
ags_file_read_start, 304
AGS_FILE_RESOLVE, 315
ags_file_rw_open, 302
ags_file_set_audio_encoding, 294
ags_file_set_audio_format, 293
ags_file_set_encoding, 292
ags_file_set_filename, 292
ags_file_set_xml_doc, 294
ags_file_str2md5, 295
ags_file_write, 303
ags_file_write_application_context, 305
ags_file_write_concurrent, 304
ags_file_write_config, 305
ags_file_write_resolve, 304
ags_file_xml_add_child, 299
ags_file_xml_get_content, 301
ags_file_xml_get_node_children, 300
ags_file_xml_get_node_name, 299
ags_file_xml_get_node_next, 300
ags_file_xml_get_node_parent, 300
ags_file_xml_get_prop, 300
ags_file_xml_get_root_element, 298
ags_file_xml_new_doc, 298
ags_file_xml_new_node, 298
ags_file_xml_set_content, 301
ags_file_xml_set_prop, 301
ags_file_xml_set_root_element, 299
AgsFileClass, 308
AgsFileError, 308
AgsFileFlags, 307
AgsFileIdRef, 317
AgsFileIdRef::resolved, 318
AgsFileIdRef:file, 317
AgsFileIdRef:node, 317
AgsFileIdRef:reference, 318
AgsFileIdRef:xpath, 318
AgsFileIdRefClass, 317
AgsFileLaunch, 331
AgsFileLaunch::start, 332
AgsFileLaunch:file, 332
AgsFileLaunch:node, 332
AgsFileLaunch:reference, 332
AgsFileLaunchClass, 332
AgsFileLink, 323
AgsFileLink:data, 323
AgsFileLink:filename, 323
AgsFileLinkClass, 323
AgsFileLookup, 327
AgsFileLookup::resolve, 328
AgsFileLookup:file, 328
AgsFileLookup:node, 328
AgsFileLookup:reference, 328
AgsFileLookupClass, 327
AgsFileResolve, 315
AGS_FRONT_CONTROLLER, 486
AGS_FRONT_CONTROLLER_CLASS, 486
ags_front_controller_do_request, 485
AGS_FRONT_CONTROLLER_GET_CLASS, 486
ags_front_controller_get_type, 487
ags_front_controller_new, 486
AgsFrontController, 487
AgsFrontController::do-request, 487
AgsFrontControllerClass, 487
AGS_FUNCTION, 42
AgsFunction, 43
AgsFunction:normalized-function, 43
AgsFunction:pivot-table, 43
AgsFunction:source-function, 44
ags_function_add_matrix, 41
AGS_FUNCTION_CLASS, 42
AGS_FUNCTION_GET_CLASS, 42
AGS_FUNCTION_GET_OBJ_MUTEX, 41
ags_function_get_type, 43
ags_function_new, 42
ags_function_remove_matrix, 41
AgsFunctionClass, 43

G

AGS_GENERIC_MAIN_LOOP, 347
AGS_GENERIC_MAIN_LOOP_CLASS, 347
AGS_GENERIC_MAIN_LOOP_DEFAULT_JIFFIE, 348
AGS_GENERIC_MAIN_LOOP_GET_CLASS, 347
ags_generic_main_loop_get_type, 348
ags_generic_main_loop_new, 347
AgsGenericMainLoop, 348
AgsGenericMainLoopClass, 348
ags_globals_get, 182
ags_globals_get_hash_table, 181
ags_globals_set, 182

I

ags_id_generator_create_uuid, 123
AGS_IS_APPLICABLE, 132
AGS_IS_APPLICABLE_INTERFACE, 132

AGS_IS_APPLICATION_CONTEXT, 139
AGS_IS_APPLICATION_CONTEXT_CLASS, 139
AGS_IS_AUTH_SECURITY_CONTEXT, 537
AGS_IS_AUTH_SECURITY_CONTEXT_CLASS, 537
AGS_IS_AUTHENTICATION, 495
AGS_IS_AUTHENTICATION_INTERFACE, 495
AGS_IS_AUTHENTICATION_MANAGER, 504
AGS_IS_AUTHENTICATION_MANAGER_CLASS, 504
AGS_IS_BUSINESS_GROUP, 509
AGS_IS_BUSINESS_GROUP_INTERFACE, 509
AGS_IS_BUSINESS_GROUP_MANAGER, 513
AGS_IS_BUSINESS_GROUP_MANAGER_CLASS, 513
AGS_IS_CERTIFICATE, 520
AGS_IS_CERTIFICATE_INTERFACE, 520
AGS_IS_CERTIFICATE_MANAGER, 525
AGS_IS_CERTIFICATE_MANAGER_CLASS, 525
AGS_IS_CONCURRENCY_PROVIDER, 339
AGS_IS_CONCURRENCY_PROVIDER_INTERFACE, 339
AGS_IS_CONFIG, 150
AGS_IS_CONFIG_CLASS, 150
AGS_IS_CONNECTABLE, 167
AGS_IS_CONNECTABLE_INTERFACE, 168
AGS_IS_CONTROLLER, 482
AGS_IS_CONTROLLER_CLASS, 482
AGS_IS_CONVERSION, 23
AGS_IS_CONVERSION_CLASS, 23
AGS_IS_COUNTABLE, 172
AGS_IS_COUNTABLE_INTERFACE, 173
AGS_IS_CURSOR, 179
AGS_IS_CURSOR_INTERFACE, 180
AGS_IS_DESTROY_WORKER, 344
AGS_IS_DESTROY_WORKER_CLASS, 344
AGS_IS_FILE, 306
AGS_IS_FILE_CLASS, 306
AGS_IS_FILE_ID_REF, 316
AGS_IS_FILE_ID_REF_CLASS, 316
AGS_IS_FILE_LAUNCH, 331
AGS_IS_FILE_LAUNCH_CLASS, 331
AGS_IS_FILE_LINK, 322
AGS_IS_FILE_LINK_CLASS, 322
AGS_IS_FILE_LOOKUP, 327
AGS_IS_FILE_LOOKUP_CLASS, 327
AGS_IS_FRONT_CONTROLLER, 486
AGS_IS_FRONT_CONTROLLER_CLASS, 487
AGS_IS_FUNCTION, 42
AGS_IS_FUNCTION_CLASS, 42
AGS_IS_GENERIC_MAIN_LOOP, 347
AGS_IS_GENERIC_MAIN_LOOP_CLASS, 347
AGS_IS_LOG, 69
AGS_IS_LOG_CLASS, 69
AGS_IS_MAIN_LOOP, 188
AGS_IS_MAIN_LOOP_INTERFACE, 188
AGS_IS_MESSAGE_DELIVERY, 355
AGS_IS_MESSAGE_DELIVERY_CLASS, 355
AGS_IS_MESSAGE_ENVELOPE, 369
AGS_IS_MESSAGE_ENVELOPE_CLASS, 369
AGS_IS_MESSAGE_QUEUE, 362
AGS_IS_MESSAGE_QUEUE_CLASS, 363
AGS_IS_MUTABLE, 212
AGS_IS_MUTABLE_INTERFACE, 212
AGS_IS_PASSWORD_STORE, 529
AGS_IS_PASSWORD_STORE_INTERFACE, 529
AGS_IS_PASSWORD_STORE_MANAGER, 534
AGS_IS_PASSWORD_STORE_MANAGER_CLASS, 534
AGS_IS_PLUGIN, 219
AGS_IS_PLUGIN_CONTROLLER, 489
AGS_IS_PLUGIN_CONTROLLER_INTERFACE, 489
AGS_IS_PLUGIN_INTERFACE, 219
AGS_IS_PORTLET, 223
AGS_IS_PORTLET_INTERFACE, 224
AGS_IS_PRIORITY, 157
AGS_IS_PRIORITY_CLASS, 157
AGS_IS_REGISTRY, 452
AGS_IS_REGISTRY_CLASS, 452
AGS_IS_RETURNABLE_THREAD, 375
AGS_IS_RETURNABLE_THREAD_CLASS, 375
AGS_IS_SECURITY_CONTEXT, 541
AGS_IS_SECURITY_CONTEXT_CLASS, 541
AGS_IS_SEEKABLE, 226
AGS_IS_SEEKABLE_INTERFACE, 226
AGS_IS_SEQUENCER, 240
AGS_IS_SEQUENCER_INTERFACE, 240
AGS_IS_SERVER, 460
AGS_IS_SERVER_APPLICATION_CONTEXT, 468
AGS_IS_SERVER_APPLICATION_CONTEXT_CLASS, 468
AGS_IS_SERVER_CLASS, 460
AGS_IS_SERVER_STATUS, 471
AGS_IS_SERVER_STATUS_CLASS, 471
AGS_IS_SERVICE_PROVIDER, 476
AGS_IS_SERVICE_PROVIDER_INTERFACE, 476
AGS_IS_SOLVER_MATRIX, 49
AGS_IS_SOLVER_MATRIX_CLASS, 49
AGS_IS_SOLVER_POLYNOMIAL, 64
AGS_IS_SOLVER_POLYNOMIAL_CLASS, 64
AGS_IS_SOLVER_VECTOR, 54
AGS_IS_SOLVER_VECTOR_CLASS, 54
AGS_IS_SOUND_SERVER, 276
AGS_IS_SOUND_SERVER_INTERFACE, 276
AGS_IS_SOUNDCARD, 262
AGS_IS_SOUNDCARD_INTERFACE, 263
AGS_IS_TACTABLE, 283
AGS_IS_TACTABLE_INTERFACE, 283
AGS_IS_TASK, 389
AGS_IS_TASK_CLASS, 389
AGS_IS_TASK_COMPLETION, 395
AGS_IS_TASK_COMPLETION_CLASS, 395
AGS_IS_TASK_LAUNCHER, 383
AGS_IS_TASK_LAUNCHER_CLASS, 383
AGS_IS_THREAD, 417
AGS_IS_THREAD_APPLICATION_CONTEXT, 428
AGS_IS_THREAD_APPLICATION_CONTEXT_CLASS, 429
AGS_IS_THREAD_CLASS, 417
AGS_IS_THREAD_POOL, 433

AGS_IS_THREAD_POOL_CLASS, 433
AGS_IS_TIMESTAMP, 441
AGS_IS_TIMESTAMP_CLASS, 441
AGS_IS_TURTLE, 109
AGS_IS_TURTLE_CLASS, 109
AGS_IS_TURTLE_MANAGER, 114
AGS_IS_TURTLE_MANAGER_CLASS, 114
AGS_IS_WORKER_THREAD, 446
AGS_IS_WORKER_THREAD_CLASS, 446
AGS_IS_XML_AUTHENTICATION, 545
AGS_IS_XML_AUTHENTICATION_CLASS, 545
AGS_IS_XML_BUSINESS_GROUP, 547
AGS_IS_XML_BUSINESS_GROUP_CLASS, 548
AGS_IS_XML_CERTIFICATE, 550
AGS_IS_XML_CERTIFICATE_CLASS, 550
AGS_IS_XML_PASSWORD_STORE, 553
AGS_IS_XML_PASSWORD_STORE_CLASS, 553

L

ags_list_util_find_type, 125
ags_list_util_get_type, 125
AgsListUtil, 125
AGS_LOG, 69
AgsLog, 70
ags_log_add_message, 68
AGS_LOG_CLASS, 69
AGS_LOG_GET_CLASS, 69
ags_log_get_instance, 69
ags_log_get_messages, 68
AGS_LOG_GET_OBJ_MUTEX, 68
ags_log_get_type, 70
ags_log_new, 69
AgsLogClass, 70
AGS_LOGIN_INFO, 497
ags_login_info_alloc, 498
ags_login_info_free, 498
ags_login_info_ref, 498
ags_login_info_unref, 498
AgsLoginInfo, 505

M

AGS_MAIN_LOOP, 188
ags_main_loop_change_frequency, 188
ags_main_loop_dec_queued_critical_region, 187
AGS_MAIN_LOOP_GET_INTERFACE, 188
ags_main_loop_get_tree_lock, 185
ags_main_loop_get_type, 189
ags_main_loop_inc_queued_critical_region, 187
AGS_MAIN_LOOP_INTERFACE, 188
ags_main_loop_is_critical_region, 186
ags_main_loop_is_syncing, 186
ags_main_loop_set_critical_region, 186
ags_main_loop_set_syncing, 185
ags_main_loop_test_queued_critical_region, 187
AgsMainLoop, 189
AgsMainLoop::change-frequency, 189
AgsMainLoopInterface, 189
ags_math_util_coefficient_to_complex, 35
ags_math_util_find_exponent_parenthesis, 31
ags_math_util_find_function, 36
ags_math_util_find_function_parenthesis, 31
ags_math_util_find_parenthesis_all, 30
ags_math_util_find_symbol, 36
ags_math_util_find_symbol_all, 36
ags_math_util_find_term_parenthesis, 32
ags_math_util_get_type, 38
ags_math_util_is_term, 37
ags_math_util_match_coefficient, 33
ags_math_util_match_exponent, 34
ags_math_util_match_function, 35
ags_math_util_match_operator, 34
ags_math_util_match_sign, 32
ags_math_util_match_symbol, 33
ags_math_util_multiply_coefficient_all, 35
ags_math_util_split_polynomial, 37
ags_math_util_split_sum, 37
AgsMathUtil, 39
AGS_MESSAGE_DELIVERY, 355
ags_message_delivery_add_message_envelope, 353
ags_message_delivery_add_message_queue, 351
AGS_MESSAGE_DELIVERY_CLASS, 356
ags_message_delivery_find_recipient, 354
ags_message_delivery_find_recipient_namespace, 352
ags_message_delivery_find_sender, 353
ags_message_delivery_find_sender_namespace, 352
AGS_MESSAGE_DELIVERY_GET_CLASS, 356
ags_message_delivery_get_instance, 355
ags_message_delivery_get_message_queue, 351
AGS_MESSAGE_DELIVERY_GET_OBJ_MUTEX, 350
ags_message_delivery_get_type, 356
ags_message_delivery_new, 355
ags_message_delivery_query_message, 354
ags_message_delivery_remove_message_envelope, 353
ags_message_delivery_remove_message_queue, 351
ags_message_delivery_set_message_queue, 350
AGS_MESSAGE_ENVELOPE, 369
AGS_MESSAGE_ENVELOPE_CLASS, 369
AGS_MESSAGE_ENVELOPE_GET_CLASS, 369
ags_message_envelope_get_doc, 367
AGS_MESSAGE_ENVELOPE_GET_OBJ_MUTEX, 366
ags_message_envelope_get_parameter, 367
ags_message_envelope_get_recipient, 367
ags_message_envelope_get_sender, 366
ags_message_envelope_get_type, 369
ags_message_envelope_new, 368
ags_message_envelope_new_with_params, 368
AGS_MESSAGE_QUEUE, 363
ags_message_queue_add_message_envelope, 360
AGS_MESSAGE_QUEUE_CLASS, 363
ags_message_queue_find_recipient, 361
ags_message_queue_find_sender, 361
AGS_MESSAGE_QUEUE_GET_CLASS, 363
ags_message_queue_get_message_envelope, 360
AGS_MESSAGE_QUEUE_GET_OBJ_MUTEX, 358

ags_message_queue_get_recipient_namespace, 359
ags_message_queue_get_sender_namespace, 358
ags_message_queue_get_type, 363
ags_message_queue_new, 362
ags_message_queue_query_message, 362
ags_message_queue_remove_message_envelope, 361
ags_message_queue_set_message_envelope, 360
ags_message_queue_set_recipient_namespace, 359
ags_message_queue_set_sender_namespace, 358
AgsMessageDelivery, 356
AgsMessageDeliveryClass, 356
AgsMessageEnvelope, 370
AgsMessageEnvelope:recipient, 370
AgsMessageEnvelope:sender, 370
AgsMessageEnvelope:xml-doc, 370
AgsMessageEnvelopeClass, 370
AgsMessageQueue, 363
AgsMessageQueue:recipient-namespace, 364
AgsMessageQueue:sender-namespace, 364
AgsMessageQueueClass, 363
AGS_MUTABLE, 212
AgsMutable, 213
AGS_MUTABLE_GET_INTERFACE, 212
ags_mutable_get_type, 213
AGS_MUTABLE_INTERFACE, 212
ags_mutable_set_muted, 212
AgsMutableInterface, 213

N

AGS_NSEC_PER_SEC, 79

P

AGS_PASSWORD_STORE, 529
ags_password_store_encrypt_password, 529
AGS_PASSWORD_STORE_GET_INTERFACE, 529
ags_password_store_get_login_name, 527
ags_password_store_get_password, 528
ags_password_store_get_type, 530
AGS_PASSWORD_STORE_INTERFACE, 529
AGS_PASSWORD_STORE_MANAGER, 534
ags_password_store_manager_add_password_store, 532
ags_password_store_manager_check_password, 533
AGS_PASSWORD_STORE_MANAGER_CLASS, 534
AGS_PASSWORD_STORE_MANAGER_GET_CLASS, 534
ags_password_store_manager_get_instance, 533
AGS_PASSWORD_STORE_MANAGER_GET_OBJ_MUTEX, 531
ags_password_store_manager_get_password_store, 532
ags_password_store_manager_get_type, 534
ags_password_store_manager_new, 533
ags_password_store_manager_remove_password_store, 532
ags_password_store_set_login_name, 527
ags_password_store_set_password, 528
AgsPasswordStore, 530
AgsPasswordStoreInterface, 530
AgsPasswordStoreManager, 534
AgsPasswordStoreManagerClass, 535
AGS_PLUGIN, 219
AgsPlugin, 220
AGS_PLUGIN_CONTROLLER, 490
ags_plugin_controller_do_request, 489
AGS_PLUGIN_CONTROLLER_GET_INTERFACE, 490
ags_plugin_controller_get_type, 490
AGS_PLUGIN_CONTROLLER_INTERFACE, 490
ags_plugin_get_build_id, 216
AGS_PLUGIN_GET_INTERFACE, 219
ags_plugin_get_name, 215
ags_plugin_get_ports, 218
ags_plugin_get_type, 220
ags_plugin_get_version, 216
ags_plugin_get_xml_type, 217
AGS_PLUGIN_INTERFACE, 220
ags_plugin_read, 218
ags_plugin_set_build_id, 217
ags_plugin_set_name, 215
ags_plugin_set_ports, 218
ags_plugin_set_version, 216
ags_plugin_set_xml_type, 217
ags_plugin_write, 219
AgsPluginController, 490
AgsPluginControllerInterface, 490
AgsPluginInterface, 220
AGS_PORTLET, 224
AgsPortlet, 224
AGS_PORTLET_GET_INTERFACE, 224
ags_portlet_get_port, 222
ags_portlet_get_type, 224
AGS_PORTLET_INTERFACE, 224
ags_portlet_list_safe_properties, 222
ags_portlet_safe_get_property, 223
ags_portlet_safe_set_property, 223
ags_portlet_set_port, 222
AgsPortletInterface, 224
AGS_PRIORITY, 157
AgsPriority, 159
AgsPriority::get-value, 159
AgsPriority::load-defaults, 160
AgsPriority::set-value, 160
AGS_PRIORITY_CLASS, 158
AGS_PRIORITY_DEFAULT_BUILD_ID, 158
AGS_PRIORITY_DEFAULT_VERSION, 158
AGS_PRIORITY_GET_CLASS, 158
ags_priority_get_instance, 157
AGS_PRIORITY_GET_OBJ_MUTEX, 155
ags_priority_get_type, 158
ags_priority_get_value, 156
AGS_PRIORITY_KEY_AUDIO, 159
AGS_PRIORITY_KEY_AUDIO_MAIN_LOOP, 159
AGS_PRIORITY_KEY_GUI_MAIN_LOOP, 159
AGS_PRIORITY_KEY_LIBAGS, 158
AGS_PRIORITY_KEY_OSC_SERVER_MAIN_LOOP, 159
AGS_PRIORITY_KEY_SERVER_MAIN_LOOP, 158
ags_priority_load_defaults, 155
ags_priority_load_from_file, 156

ags_priority_new, 157
AGS_PRIORITY_RT_THREAD, 158
ags_priority_set_value, 156
AgsPriorityClass, 159

R

ags_regcomp, 71
ags_regexec, 72
AGS_REGISTRY, 453
AgsRegistry, 454
AgsRegistry:server, 454
ags_registry_add_entry, 452
AGS_REGISTRY_CLASS, 453
ags_registry_entry_alloc, 451
ags_registry_entry_free, 451
ags_registry_find_entry, 452
AGS_REGISTRY_GET_CLASS, 453
AGS_REGISTRY_GET_OBJ_MUTEX, 451
ags_registry_get_type, 453
ags_registry_new, 452
AgsRegistryClass, 454
AgsRegistryEntry, 453
AGS_RETURNABLE_THREAD, 375
AGS_RETURNABLE_THREAD_CLASS, 375
ags_returnable_thread_connect_safe_run, 374
AGS_RETURNABLE_THREAD_DEFAULT_JIFFIE, 375
ags_returnable_thread_disconnect_safe_run, 374
ags_returnable_thread_flags_get_type, 375
AGS_RETURNABLE_THREAD_GET_CLASS, 375
AGS_RETURNABLE_THREAD_GET_RESET_MUTEX, 372
ags_returnable_thread_get_type, 375
ags_returnable_thread_new, 374
ags_returnable_thread_safe_run, 374
ags_returnable_thread_set_flags, 373
ags_returnable_thread_test_flags, 373
ags_returnable_thread_unset_flags, 373
AgsReturnableThread, 376
AgsReturnableThread::safe-run, 377
AgsReturnableThread:thread-pool, 376
AgsReturnableThreadCallback, 372
AgsReturnableThreadClass, 376
AgsReturnableThreadFlags, 375

S

AGS_SECURITY_CONTEXT, 541
ags_security_context_add_server_context, 539
AGS_SECURITY_CONTEXT_CLASS, 541
ags_security_context_get_business_group, 539
AGS_SECURITY_CONTEXT_GET_CLASS, 541
AGS_SECURITY_CONTEXT_GET_OBJ_MUTEX, 538
ags_security_context_get_server_context, 540
ags_security_context_get_type, 541
ags_security_context_mode_get_type, 541
ags_security_context_new, 540
ags_security_context_parse_business_group, 539
ags_security_context_remove_server_context, 540
AgsSecurityContext, 542
AgsSecurityContext:certs, 543
AgsSecurityContextClass, 543
AgsSecurityContextMode, 541
ags_seek_type_get_type, 227
AGS_SEEKABLE, 227
AgsSeekable, 228
AgsSeekable::seek, 228
AGS_SEEKABLE_GET_INTERFACE, 227
ags_seekable_get_type, 227
AGS_SEEKABLE_INTERFACE, 227
ags_seekable_seek, 226
AgsSeekableInterface, 228
AgsSeekType, 227
AGS_SEQUENCER, 240
AgsSequencer, 242
AgsSequencer::offset-changed, 243
AgsSequencer::tic, 244
AGS_SEQUENCER_DEFAULT_BPM, 241
AGS_SEQUENCER_DEFAULT_BUFFER_SIZE, 241
AGS_SEQUENCER_DEFAULT_DELAY, 242
AGS_SEQUENCER_DEFAULT_DELAY_FACTOR, 241
AGS_SEQUENCER_DEFAULT_DEVICE, 241
AGS_SEQUENCER_DEFAULT_JIFFIE, 241
AGS_SEQUENCER_DEFAULT_LATENCY, 242
AGS_SEQUENCER_DEFAULT_PERIOD, 242
AGS_SEQUENCER_DEFAULT_SAMPLERATE, 241
AGS_SEQUENCER_DEFAULT_SCALE, 242
AGS_SEQUENCER_DEFAULT_TACT, 241
AGS_SEQUENCER_DEFAULT_TACT_JIFFIE, 241
AGS_SEQUENCER_DEFAULT_TACTRATE, 241
ags_sequencer_get_bpm, 237
ags_sequencer_get_buffer, 236
ags_sequencer_get_delay_factor, 238
ags_sequencer_get_device, 232
AGS_SEQUENCER_GET_INTERFACE, 240
ags_sequencer_get_next_buffer, 236
ags_sequencer_get_note_offset, 239
ags_sequencer_get_start_note_offset, 239
ags_sequencer_get_type, 240
AGS_SEQUENCER_INTERFACE, 240
ags_sequencer_is_playing, 233
ags_sequencer_is_recording, 233
ags_sequencer_is_starting, 233
ags_sequencer_list_cards, 232
ags_sequencer_lock_buffer, 237
AGS_SEQUENCER_MAX_MIDI_CHANNELS, 240
AGS_SEQUENCER_MAX_MIDI_KEYS, 241
ags_sequencer_offset_changed, 236
ags_sequencer_play, 234
ags_sequencer_play_init, 234
ags_sequencer_record, 235
ags_sequencer_record_init, 234
ags_sequencer_set_bpm, 237
ags_sequencer_set_delay_factor, 238
ags_sequencer_set_device, 232
ags_sequencer_set_note_offset, 239

ags_sequencer_set_start_note_offset, 238
ags_sequencer_stop, 235
ags_sequencer_tic, 235
ags_sequencer_unlock_buffer, 237
AgsSequencerInterface, 242
AGS_SERVER, 460
AgsServer, 463
AgsServer::listen, 465
AgsServer::start, 465
AgsServer::stop, 466
AgsServer:controller, 463
AgsServer:domain, 463
AgsServer:front-controller, 463
AgsServer:ip4, 464
AgsServer:ip6, 464
AgsServer:path, 464
AgsServer:realm, 464
AgsServer:server-port, 465
ags_server_add_controller, 458
AGS_SERVER_APPLICATION_CONTEXT, 468
AGS_SERVER_APPLICATION_CONTEXT_CLASS, 468
AGS_SERVER_APPLICATION_CONTEXT_GET_CLASS, 468
ags_server_application_context_get_type, 468
ags_server_application_context_new, 468
AGS_SERVER_BUILD_ID, 469
AGS_SERVER_CLASS, 460
AGS_SERVER_DEFAULT_AUTH_MODULE, 461
AGS_SERVER_DEFAULT_BACKLOG, 461
AGS_SERVER_DEFAULT_DOMAIN, 461
AGS_SERVER_DEFAULT_INET4_ADDRESS, 461
AGS_SERVER_DEFAULT_INET6_ADDRESS, 461
AGS_SERVER_DEFAULT_SERVER_PORT, 460
ags_server_flags_get_type, 460
AGS_SERVER_GET_CLASS, 460
AGS_SERVER_GET_OBJ_MUTEX, 456
ags_server_get_type, 460
ags_server_info_alloc, 458
ags_server_listen, 459
ags_server_new, 459
ags_server_remove_controller, 458
ags_server_set_flags, 457
ags_server_start, 459
AGS_SERVER_STATUS, 471
AGS_SERVER_STATUS_CLASS, 471
AGS_SERVER_STATUS_GET_CLASS, 471
AGS_SERVER_STATUS_GET_OBJ_MUTEX, 471
ags_server_status_get_type, 472
ags_server_status_new, 471
ags_server_stop, 459
ags_server_test_flags, 457
ags_server_unset_flags, 457
AGS_SERVER_VERSION, 469
AgsServerApplicationContext, 469
AgsServerApplicationContextClass, 469
AgsServerClass, 463
AgsServerFlags, 461
AgsServerInfo, 462
AgsServerStatus, 472
AgsServerStatusClass, 472
AGS_SERVICE_PROVIDER, 476
AGS_SERVICE_PROVIDER_GET_INTERFACE, 476
ags_service_provider_get_registry, 475
ags_service_provider_get_server, 475
ags_service_provider_get_type, 476
AGS_SERVICE_PROVIDER_INTERFACE, 476
ags_service_provider_is_operating, 474
ags_service_provider_server_status, 474
ags_service_provider_set_registry, 474
ags_service_provider_set_server, 475
AgsServiceProvider, 477
AgsServiceProviderInterface, 477
AGS_SOLVER_MATRIX, 49
AGS_SOLVER_MATRIX_CLASS, 49
ags_solver_matrix_eliminate, 48
AGS_SOLVER_MATRIX_GET_CLASS, 49
ags_solver_matrix_get_column_count, 47
AGS_SOLVER_MATRIX_GET_OBJ_MUTEX, 46
ags_solver_matrix_get_row_count, 47
ags_solver_matrix_get_source_function, 46
ags_solver_matrix_get_type, 49
ags_solver_matrix_insert_vector, 48
ags_solver_matrix_new, 49
ags_solver_matrix_parse, 47
ags_solver_matrix_remove_vector, 48
ags_solver_matrix_set_source_function, 46
AGS_SOLVER_POLYNOMIAL, 64
ags_solver_polynomial_add, 61
AGS_SOLVER_POLYNOMIAL_CLASS, 64
ags_solver_polynomial_divide, 62
AGS_SOLVER_POLYNOMIAL_ERROR, 65
ags_solver_polynomial_error_quark, 57
ags_solver_polynomial_extract_root, 63
AGS_SOLVER_POLYNOMIAL_GET_CLASS, 64
ags_solver_polynomial_get_coefficient, 59
ags_solver_polynomial_get_coefficient_value, 60
ags_solver_polynomial_get_exponent, 59
ags_solver_polynomial_get_exponent_value, 60
AGS_SOLVER_POLYNOMIAL_GET_OBJ_MUTEX, 57
ags_solver_polynomial_get_polynomial, 58
ags_solver_polynomial_get_symbol, 59
ags_solver_polynomial_get_type, 64
ags_solver_polynomial_multiply, 62
ags_solver_polynomial_new, 64
ags_solver_polynomial_parse, 61
ags_solver_polynomial_raise_power, 63
ags_solver_polynomial_set_coefficient, 58
ags_solver_polynomial_set_coefficient_value, 60
ags_solver_polynomial_set_polynomial, 58
ags_solver_polynomial_subtract, 62
ags_solver_polynomial_update, 61
AGS_SOLVER_VECTOR, 54
AGS_SOLVER_VECTOR_CLASS, 54
AGS_SOLVER_VECTOR_GET_CLASS, 54

AGS_SOLVER_VECTOR_GET_OBJ_MUTEX, 52
ags_solver_vector_get_polynomial_count, 53
ags_solver_vector_get_source_polynomial, 52
ags_solver_vector_get_type, 55
ags_solver_vector_insert_polynomial, 53
ags_solver_vector_new, 54
ags_solver_vector_remove_polynomial, 53
ags_solver_vector_set_source_polynomial, 52
AgsSolverMatrix, 50
AgsSolverMatrix:column-count, 50
AgsSolverMatrix:row-count, 50
AgsSolverMatrix:source-function, 50
AgsSolverMatrixClass, 50
AgsSolverPolynomial, 65
AgsSolverPolynomial:coefficient, 65
AgsSolverPolynomial:coefficient-value, 66
AgsSolverPolynomial:exponent, 66
AgsSolverPolynomial:exponent-value, 66
AgsSolverPolynomial:polynomial, 66
AgsSolverPolynomial:symbol, 66
AgsSolverPolynomialClass, 65
AgsSolverPolynomialError, 65
AgsSolverVector, 55
AgsSolverVector:polynomial-count, 55
AgsSolverVector:source-polynomial, 55
AgsSolverVectorClass, 55
AGS_SOUND_SERVER, 276
AGS_SOUND_SERVER_GET_INTERFACE, 276
ags_sound_server_get_ports, 273
ags_sound_server_get_sequencer, 274
ags_sound_server_get_soundcard, 273
ags_sound_server_get_type, 277
ags_sound_server_get_url, 272
AGS_SOUND_SERVER_INTERFACE, 276
ags_sound_server_register_sequencer, 275
ags_sound_server_register_soundcard, 275
ags_sound_server_set_ports, 273
ags_sound_server_set_sequencer, 274
ags_sound_server_set_soundcard, 273
ags_sound_server_set_url, 272
ags_sound_server_unregister_sequencer, 276
ags_sound_server_unregister_soundcard, 275
AGS_SOUNDCARD, 263
AgsSoundcard, 267
AgsSoundcard::offset-changed, 269
AgsSoundcard::stop, 270
AgsSoundcard::tic, 270
ags_soundcard_capability_get_type, 263
AGS_SOUNDCARD_DEFAULT_BPM, 265
AGS_SOUNDCARD_DEFAULT_BUFFER_SIZE, 264
AGS_SOUNDCARD_DEFAULT_DELAY, 265
AGS_SOUNDCARD_DEFAULT_DELAY_FACTOR, 265
AGS_SOUNDCARD_DEFAULT_DSP_CHANNELS, 263
AGS_SOUNDCARD_DEFAULT_FORMAT, 264
AGS_SOUNDCARD_DEFAULT_JIFFIE, 265
AGS_SOUNDCARD_DEFAULT_LOOP_LEFT, 266
AGS_SOUNDCARD_DEFAULT_LOOP_RIGHT, 266
AGS_SOUNDCARD_DEFAULT_OVERCLOCK, 266
AGS_SOUNDCARD_DEFAULT_PCM_CHANNELS, 264
AGS_SOUNDCARD_DEFAULT_PERIOD, 265
AGS_SOUNDCARD_DEFAULT_SAMPLERATE, 264
AGS_SOUNDCARD_DEFAULT_SCALE, 265
AGS_SOUNDCARD_DEFAULT_SUB_BLOCK_COUNT, 266
AGS_SOUNDCARD_DEFAULT_TACT, 265
AGS_SOUNDCARD_DEFAULT_TACT_JIFFIE, 265
AGS_SOUNDCARD_DEFAULT_TACTRATE, 265
ags_soundcard_format_get_type, 263
ags_soundcard_get_absolute_delay, 257
ags_soundcard_get_attack, 258
ags_soundcard_get_bpm, 256
ags_soundcard_get_buffer, 254
ags_soundcard_get_capability, 250
ags_soundcard_get_delay, 257
ags_soundcard_get_delay_counter, 258
ags_soundcard_get_delay_factor, 257
ags_soundcard_get_device, 248
AGS_SOUNDCARD_GET_INTERFACE, 263
ags_soundcard_get_loop, 261
ags_soundcard_get_loop_offset, 261
ags_soundcard_get_next_buffer, 254
ags_soundcard_get_note_offset, 259
ags_soundcard_get_note_offset_absolute, 260
ags_soundcard_get_presets, 248
ags_soundcard_get_prev_buffer, 255
ags_soundcard_get_start_note_offset, 259
ags_soundcard_get_sub_block_count, 261
ags_soundcard_get_type, 263
ags_soundcard_get_uptime, 251
ags_soundcard_helper_config_get_buffer_size, 127
ags_soundcard_helper_config_get_dsp_channels, 126
ags_soundcard_helper_config_get_format, 128
ags_soundcard_helper_config_get_pcm_channels, 127
ags_soundcard_helper_config_get_samplerate, 127
AGS_SOUNDCARD_INTERFACE, 263
ags_soundcard_is_available, 250
ags_soundcard_is_playing, 251
ags_soundcard_is_recording, 251
ags_soundcard_is_starting, 250
ags_soundcard_list_cards, 249
ags_soundcard_lock_buffer, 255
AGS_SOUNDCARD_MAX_BUFFER_SIZE, 265
AGS_SOUNDCARD_MAX_DSP_CHANNELS, 264
AGS_SOUNDCARD_MAX_PCM_CHANNELS, 264
AGS_SOUNDCARD_MAX_SAMPLERATE, 264
AGS_SOUNDCARD_MIN_BUFFER_SIZE, 264
AGS_SOUNDCARD_MIN_DSP_CHANNELS, 263
AGS_SOUNDCARD_MIN_PCM_CHANNELS, 264
AGS_SOUNDCARD_MIN_SAMPLERATE, 264
ags_soundcard_offset_changed, 254
ags_soundcard_pcm_info, 249
ags_soundcard_play, 252
ags_soundcard_play_init, 252
ags_soundcard_record, 253

ags_soundcard_record_init, 252
ags_soundcard_set_bpm, 256
ags_soundcard_set_delay_factor, 256
ags_soundcard_set_device, 247
ags_soundcard_set_loop, 260
ags_soundcard_set_note_offset, 259
ags_soundcard_set_note_offset_absolute, 260
ags_soundcard_set_presets, 248
ags_soundcard_set_start_note_offset, 258
ags_soundcard_stop, 253
ags_soundcard_tic, 253
ags_soundcard_trylock_sub_block, 262
ags_soundcard_unlock_buffer, 255
ags_soundcard_unlock_sub_block, 262
AgsSoundcardCapability, 267
AgsSoundcardFormat, 266
AgsSoundcardInterface, 268
AgsSoundServer, 277
AgsSoundServerInterface, 277
ags_string_util_escape_single_quote, 74
ags_string_util_get_type, 76
AgsStringUtil, 76
ags_strv_contains, 74
ags_strv_equal, 75
ags_strv_index, 75
ags_strv_length, 74
AGS_SUBSCRIPT_0, 38
AGS_SUBSCRIPT_1, 38
AGS_SUBSCRIPT_2, 38
AGS_SUBSCRIPT_3, 38
AGS_SUBSCRIPT_4, 38
AGS_SUBSCRIPT_5, 39
AGS_SUBSCRIPT_6, 39
AGS_SUBSCRIPT_7, 39
AGS_SUBSCRIPT_8, 39
AGS_SUBSCRIPT_9, 39
AGS_SYMBOLIC_COMPLEX_UNIT, 38
AGS_SYMBOLIC_EULER, 38
AGS_SYMBOLIC_INFINIT, 38
AGS_SYMBOLIC_PI, 38

T

AGS_TACTABLE, 283
AgsTactable, 284
AgsTactable::change-bpm, 284
AgsTactable::change-midi-duration, 285
AgsTactable::change-notation-duration, 285
AgsTactable::change-sequencer-duration, 286
AgsTactable::change-tact, 286
AgsTactable::change-wave-duration, 286
ags_tactable_change_bpm, 282
ags_tactable_change_midi_duration, 282
ags_tactable_change_notation_duration, 281
ags_tactable_change_sequencer_duration, 281
ags_tactable_change_tact, 283
ags_tactable_change_wave_duration, 282
ags_tactable_get_bpm, 280
AGS_TACTABLE_GET_INTERFACE, 283
ags_tactable_get_midi_duration, 280
ags_tactable_get_notation_duration, 279
ags_tactable_get_sequencer_duration, 279
ags_tactable_get_tact, 281
ags_tactable_get_type, 284
ags_tactable_get_wave_duration, 280
AGS_TACTABLE_INTERFACE, 283
AgsTactableInterface, 284
AGS_TASK, 389
AgsTask, 390
AgsTask::failure, 391
AgsTask::launch, 391
AgsTask:task-launcher, 391
AGS_TASK_CLASS, 389
AGS_TASK_COMPLETION, 395
AGS_TASK_COMPLETION_CLASS, 395
ags_task_completion_complete, 394
ags_task_completion_flags_get_type, 396
AGS_TASK_COMPLETION_GET_CLASS, 396
AGS_TASK_COMPLETION_GET_OBJ_MUTEX, 393
ags_task_completion_get_type, 396
ags_task_completion_new, 395
ags_task_completion_set_flags, 394
ags_task_completion_test_flags, 393
ags_task_completion_unset_flags, 394
ags_task_failure, 389
AGS_TASK_GET_CLASS, 390
AGS_TASK_GET_OBJ_MUTEX, 387
ags_task_get_type, 390
ags_task_launch, 388
AGS_TASK_LAUNCHER, 383
ags_task_launcher_add_cyclic_task, 381
ags_task_launcher_add_task, 381
ags_task_launcher_add_task_all, 381
ags_task_launcher_attach, 380
AGS_TASK_LAUNCHER_CLASS, 383
AGS_TASK_LAUNCHER_GET_CLASS, 383
AGS_TASK_LAUNCHER_GET_OBJ_MUTEX, 379
ags_task_launcher_get_type, 383
ags_task_launcher_new, 383
ags_task_launcher_remove_cyclic_task, 382
ags_task_launcher_run, 382
ags_task_launcher_set_flags, 380
ags_task_launcher_sync_run, 382
ags_task_launcher_test_flags, 379
ags_task_launcher_unset_flags, 380
ags_task_new, 389
ags_task_set_flags, 388
ags_task_test_flags, 387
ags_task_unset_flags, 388
AgsTaskClass, 390
AgsTaskCompletion, 397
AgsTaskCompletion::complete, 397
AgsTaskCompletion:task, 397
AgsTaskCompletionClass, 397
AgsTaskCompletionFlags, 396

AgsTaskFlags, 390
AgsTaskLauncher, 384
AgsTaskLauncher::run, 384
AgsTaskLauncher:cyclic-task, 384
AgsTaskLauncher:task, 384
AgsTaskLauncherClass, 384
AGS_THREAD, 417
AgsThread, 423
AgsThread::clock, 424
AgsThread::run, 425
AgsThread::start, 425
AgsThread::stop, 426
AgsThread:delay, 424
AgsThread:frequency, 424
AgsThread:max-precision, 424
ags_thread_add_child, 413
ags_thread_add_child_extended, 414
ags_thread_add_start_queue, 415
ags_thread_add_start_queue_all, 415
AGS_THREAD_APPLICATION_CONTEXT, 429
AGS_THREAD_APPLICATION_CONTEXT_CLASS, 429
ags_thread_application_context_flags_get_type, 429
AGS_THREAD_APPLICATION_CONTEXT_GET_CLASS, 429
ags_thread_application_context_get_type, 429
ags_thread_application_context_new, 428
ags_thread_application_context_register_types, 428
AGS_THREAD_BUILD_ID, 429
ags_thread_children, 411
AGS_THREAD_CLASS, 417
ags_thread_clear_status_flags, 405
ags_thread_clear_sync_tic_flags, 406
ags_thread_clock, 415
AGS_THREAD_DEFAULT_ATTACK, 418
AGS_THREAD_DEFAULT_JIFFIE, 418
AGS_THREAD_DEFAULT_MAX_PRECISION, 418
AGS_THREAD_DEFAULT_VERSION, 429
ags_thread_find_type, 409
ags_thread_first, 411
ags_thread_flags_get_type, 417
AGS_THREAD_GET_CLASS, 417
ags_thread_get_current_sync_tic, 406
ags_thread_get_delay, 407
ags_thread_get_frequency, 408
ags_thread_get_max_precision, 408
AGS_THREAD_GET_OBJ_MUTEX, 402
AGS_THREAD_GET_START_COND, 402
AGS_THREAD_GET_START_MUTEX, 402
AGS_THREAD_GET_TIC_COND, 402
AGS_THREAD_GET_TIC_MUTEX, 402
ags_thread_get_toplevel, 411
ags_thread_get_type, 417
AGS_THREAD_GET_WAIT_COND, 402
AGS_THREAD_GET_WAIT_MUTEX, 402
ags_thread_global_get_use_sync_counter, 402
AGS_THREAD_HERTZ_JIFFIE, 418
ags_thread_is_current_ready, 414
ags_thread_is_tree_ready_recursive, 414
ags_thread_last, 412
ags_thread_lock, 412
AGS_THREAD_MAX_PRECISION, 418
ags_thread_new, 417
ags_thread_next, 410
ags_thread_parent, 409
AGS_THREAD_POOL, 433
AGS_THREAD_POOL_CLASS, 434
AGS_THREAD_POOL_DEFAULT_MAX_THREADS, 434
AGS_THREAD_POOL_DEFAULT_MAX_UNUSED_THREADS 434
ags_thread_pool_flags_get_type, 434
AGS_THREAD_POOL_GET_CLASS, 434
ags_thread_pool_get_type, 434
ags_thread_pool_new, 433
ags_thread_pool_pull, 432
ags_thread_pool_start, 433
ags_thread_prepare_current_sync, 414
ags_thread_prepare_tree_sync_recursive, 414
ags_thread_prev, 410
ags_thread_remove_child, 413
ags_thread_run, 416
ags_thread_self, 409
ags_thread_set_current_sync, 414
ags_thread_set_current_sync_tic, 406
ags_thread_set_delay, 407
ags_thread_set_flags, 403
ags_thread_set_frequency, 407
ags_thread_set_max_precision, 408
ags_thread_set_status_flags, 404
ags_thread_set_sync_tic_flags, 405
ags_thread_set_tree_sync_recursive, 415
ags_thread_start, 416
ags_thread_status_flags_get_type, 418
ags_thread_stop, 416
ags_thread_sync_tic_flags_get_type, 418
ags_thread_test_flags, 403
ags_thread_test_status_flags, 404
ags_thread_test_sync_tic_flags, 405
AGS_THREAD_TOLERANCE, 418
ags_thread_trylock, 412
ags_thread_unlock, 413
ags_thread_unset_flags, 403
ags_thread_unset_status_flags, 404
ags_thread_unset_sync_tic_flags, 406
AGS_THREAD_YIELD_JIFFIE, 418
AgsThreadApplicationContext, 430
AgsThreadApplicationContext:thread-pool, 430
AgsThreadApplicationContextClass, 430
AgsThreadApplicationContextFlags, 429
AgsThreadClass, 423
AgsThreadFlags, 418
AgsThreadPool, 435
AgsThreadPool::start, 436
AgsThreadPool:max-threads, 435
AgsThreadPool:max-unused-threads, 436

AgsThreadPoolClass, 435
AgsThreadPoolFlags, 434
AgsThreadStatusFlags, 419
AgsThreadSyncTicFlags, 421
ags_time_get_uptime_from_offset, 77
ags_time_timeout_expired, 78
AGS_TIME_ZERO, 78
AGS_TIMESTAMP, 441
AgsTimestamp, 442
AGS_TIMESTAMP_CLASS, 441
ags_timestamp_flags_get_type, 441
ags_timestamp_get_ags_offset, 440
AGS_TIMESTAMP_GET_CLASS, 441
AGS_TIMESTAMP_GET_OBJ_MUTEX, 438
ags_timestamp_get_type, 441
ags_timestamp_get_unix_time, 439
ags_timestamp_new, 440
ags_timestamp_set_ags_offset, 440
ags_timestamp_set_flags, 438
ags_timestamp_set_unix_time, 439
ags_timestamp_test_flags, 438
ags_timestamp_unset_flags, 439
AgsTimestampClass, 442
AgsTimestampFlags, 441
AGS_TURTLE, 109
AgsTurtle, 110
AgsTurtle:filename, 111
AgsTurtle:xml-doc, 111
AGS_TURTLE_BOOLEAN_LITERAL_FALSE, 110
AGS_TURTLE_BOOLEAN_LITERAL_TRUE, 110
AGS_TURTLE_CLASS, 109
AGS_TURTLE_DEFAULT_ENCODING, 110
AGS_TURTLE_DEFAULT_VERSION, 110
ags_turtle_find_xpath, 107
ags_turtle_find_xpath_with_context_node, 107
ags_turtle_flags_get_type, 109
AGS_TURTLE_GET_CLASS, 109
AGS_TURTLE_GET_OBJ_MUTEX, 82
ags_turtle_get_type, 109
ags_turtle_load, 108
AGS_TURTLE_MANAGER, 114
ags_turtle_manager_add, 113
AGS_TURTLE_MANAGER_CLASS, 114
ags_turtle_manager_find, 113
AGS_TURTLE_MANAGER_GET_CLASS, 115
ags_turtle_manager_get_instance, 114
AGS_TURTLE_MANAGER_GET_OBJ_MUTEX, 113
ags_turtle_manager_get_type, 115
ags_turtle_manager_new, 114
ags_turtle_match_anon, 102
ags_turtle_match_blank_node_label, 86
ags_turtle_match_boolean, 89
ags_turtle_match_decimal, 90
ags_turtle_match_double, 90
ags_turtle_match_echar, 101
ags_turtle_match_exponent, 91
ags_turtle_match_hex, 106
ags_turtle_match_integer, 89
ags_turtle_match_iriref, 84
ags_turtle_match_langtag, 86
ags_turtle_match_percent, 106
ags_turtle_match_plx, 105
ags_turtle_match_pn_chars, 104
ags_turtle_match_pn_chars_base, 103
ags_turtle_match_pn_chars_u, 103
ags_turtle_match_pn_local, 105
ags_turtle_match_pn_local_esc, 107
ags_turtle_match_pn_prefix, 104
ags_turtle_match_pname_ln, 85
ags_turtle_match_pname_ns, 85
ags_turtle_match_string, 91
ags_turtle_match_string_literal_long_quote, 95
ags_turtle_match_string_literal_long_single_quote, 95
ags_turtle_match_string_literal_quote, 94
ags_turtle_match_string_literal_single_quote, 94
ags_turtle_match_uchar, 101
ags_turtle_match_ws, 102
ags_turtle_new, 108
ags_turtle_read_anon, 97
ags_turtle_read_blank_node_label, 83
ags_turtle_read_boolean, 87
ags_turtle_read_decimal, 87
ags_turtle_read_double, 88
ags_turtle_read_echar, 96
ags_turtle_read_exponent, 88
ags_turtle_read_hex, 100
ags_turtle_read_integer, 87
ags_turtle_read_iriref, 82
ags_turtle_read_langtag, 84
ags_turtle_read_percent, 100
ags_turtle_read_plx, 99
ags_turtle_read_pn_chars, 98
ags_turtle_read_pn_chars_base, 97
ags_turtle_read_pn_chars_u, 98
ags_turtle_read_pn_local, 99
ags_turtle_read_pn_local_esc, 100
ags_turtle_read_pn_prefix, 98
ags_turtle_read_pname_ln, 83
ags_turtle_read_pname_ns, 83
ags_turtle_read_string, 91
ags_turtle_read_string_literal_long_quote, 93
ags_turtle_read_string_literal_long_single_quote, 93
ags_turtle_read_string_literal_quote, 92
ags_turtle_read_string_literal_single_quote, 92
ags_turtle_read_uchar, 96
ags_turtle_read_ws, 96
ags_turtle_string_convert, 108
AgsTurtleClass, 111
AgsTurtleFlags, 110
AgsTurtleManager, 115
AgsTurtleManagerClass, 115
AGS_TYPE_APPLICABLE, 132
AGS_TYPE_APPLICATION_CONTEXT, 140
AGS_TYPE_APPLICATION_CONTEXT_FLAGS, 140

AGS_TYPE_AUTH_SECURITY_CONTEXT, 537
AGS_TYPE_AUTHENTICATION, 495
AGS_TYPE_AUTHENTICATION_MANAGER, 505
AGS_TYPE_BUFFER_UTIL, 16
AGS_TYPE_BUSINESS_GROUP, 509
AGS_TYPE_BUSINESS_GROUP_MANAGER, 514
AGS_TYPE_BYTE_ORDER, 28
AGS_TYPE_CERTIFICATE, 520
AGS_TYPE_CERTIFICATE_MANAGER, 525
AGS_TYPE_COMPLEX, 20
AGS_TYPE_CONCURRENCY_PROVIDER, 340
AGS_TYPE_CONFIG, 151
AGS_TYPE_CONNECTABLE, 168
AGS_TYPE_CONTROLLER, 484
AGS_TYPE_CONVERSION, 23
AGS_TYPE_COUNTABLE, 173
AGS_TYPE_CURSOR, 180
AGS_TYPE_DESTROY_UTIL, 122
AGS_TYPE_DESTROY_WORKER, 345
AGS_TYPE_FILE, 308
AGS_TYPE_FILE_FLAGS, 308
AGS_TYPE_FILE_ID_REF, 317
AGS_TYPE_FILE_LAUNCH, 331
AGS_TYPE_FILE_LINK, 323
AGS_TYPE_FILE_LOOKUP, 327
AGS_TYPE_FRONT_CONTROLLER, 487
AGS_TYPE_FUNCTION, 43
AGS_TYPE_GENERIC_MAIN_LOOP, 348
AGS_TYPE_LIST_UTIL, 125
AGS_TYPE_LOG, 70
AGS_TYPE_MAIN_LOOP, 189
AGS_TYPE_MATH_UTIL, 39
AGS_TYPE_MESSAGE_DELIVERY, 356
AGS_TYPE_MESSAGE_ENVELOPE, 369
AGS_TYPE_MESSAGE_QUEUE, 363
AGS_TYPE_MUTABLE, 213
AGS_TYPE_PASSWORD_STORE, 530
AGS_TYPE_PASSWORD_STORE_MANAGER, 534
AGS_TYPE_PLUGIN, 220
AGS_TYPE_PLUGIN_CONTROLLER, 490
AGS_TYPE_PORTLET, 224
AGS_TYPE_PRIORITY, 159
AGS_TYPE_PRIORITY_FLAGS, 158
AGS_TYPE_REGISTRY, 454
AGS_TYPE_REGISTRY_FLAGS, 453
AGS_TYPE_RETURNABLE_THREAD, 376
AGS_TYPE_RETURNABLE_THREAD_FLAGS, 376
AGS_TYPE_SECURITY_CONTEXT, 542
AGS_TYPE_SECURITY_CONTEXT_MODE, 542
AGS_TYPE_SEEK_TYPE, 228
AGS_TYPE_SEEKABLE, 228
AGS_TYPE_SEQUENCER, 242
AGS_TYPE_SERVER, 462
AGS_TYPE_SERVER_APPLICATION_CONTEXT, 469
AGS_TYPE_SERVER_FLAGS, 462
AGS_TYPE_SERVER_STATUS, 472
AGS_TYPE_SERVICE_PROVIDER, 476
AGS_TYPE_SOLVER_MATRIX, 49
AGS_TYPE_SOLVER_POLYNOMIAL, 65
AGS_TYPE_SOLVER_VECTOR, 55
AGS_TYPE_SOUND_SERVER, 277
AGS_TYPE_SOUNDCARD, 267
AGS_TYPE_SOUNDCARD_CAPABILITY, 267
AGS_TYPE_SOUNDCARD_FORMAT, 267
AGS_TYPE_STRING_UTIL, 76
AGS_TYPE_TACTABLE, 284
AGS_TYPE_TASK, 390
AGS_TYPE_TASK_COMPLETION, 397
AGS_TYPE_TASK_COMPLETION_FLAGS, 397
AGS_TYPE_TASK_FLAGS, 390
AGS_TYPE_TASK_LAUNCHER, 384
AGS_TYPE_TASK_LAUNCHER_FLAGS, 383
AGS_TYPE_THREAD, 423
AGS_TYPE_THREAD_APPLICATION_CONTEXT, 430
AGS_TYPE_THREAD_APPLICATION_CONTEXT_FLAGS, 430
AGS_TYPE_THREAD_FLAGS, 423
AGS_TYPE_THREAD_POOL, 435
AGS_TYPE_THREAD_POOL_FLAGS, 435
AGS_TYPE_THREAD_STATUS_FLAGS, 423
AGS_TYPE_THREAD_SYNC_TIC_FLAGS, 423
AGS_TYPE_TIMESTAMP, 442
AGS_TYPE_TIMESTAMP_FLAGS, 442
AGS_TYPE_TURTLE, 110
AGS_TYPE_TURTLE_FLAGS, 110
AGS_TYPE_TURTLE_MANAGER, 115
AGS_TYPE_UUID, 119
AGS_TYPE_WORKER_THREAD, 447
AGS_TYPE_WORKER_THREAD_STATUS_FLAGS, 447
AGS_TYPE_XML_AUTHENTICATION, 546
AGS_TYPE_XML_BUSINESS_GROUP, 548
AGS_TYPE_XML_CERTIFICATE, 551
AGS_TYPE_XML_PASSWORD_STORE, 554

U

AGS_USEC_PER_SEC, 78
AgsUUID, 119
ags_uuid_alloc, 117
ags_uuid_compare, 118
ags_uuid_copy, 117
AGS_UUID_DEFAULT_LENGTH, 119
ags_uuid_free, 117
ags_uuid_from_string, 119
ags_uuid_generate, 118
ags_uuid_get_type, 119
AGS_UUID_STRING_DEFAULT_LENGTH, 119
ags_uuid_to_string, 118

V

AGS_VERSION, 140

W

ags_woker_thread_do_poll_loop, 445
AGS_WORKER_THREAD, 446

AGS_WORKER_THREAD_CLASS, 446
AGS_WORKER_THREAD_DEFAULT_JIFFIE, 447
ags_worker_thread_do_poll, 445
AGS_WORKER_THREAD_GET_CLASS, 446
ags_worker_thread_get_type, 446
ags_worker_thread_new, 446
ags_worker_thread_set_status_flags, 445
ags_worker_thread_status_flags_get_type, 447
ags_worker_thread_test_status_flags, 444
ags_worker_thread_unset_status_flags, 445
AgsWorkerThread, 447
AgsWorkerThread::do-poll, 448
AgsWorkerThreadClass, 447
AgsWorkerThreadStatusFlags, 447

X

AGS_XML_AUTHENTICATION, 545
AGS_XML_AUTHENTICATION_CLASS, 545
ags_xml_authentication_find_user_uuid, 544
AGS_XML_AUTHENTICATION_GET_CLASS, 545
AGS_XML_AUTHENTICATION_GET_OBJ_MUTEX, 544
ags_xml_authentication_get_type, 545
ags_xml_authentication_new, 545
ags_xml_authentication_open_filename, 544
AGS_XML_BUSINESS_GROUP, 548
AGS_XML_BUSINESS_GROUP_CLASS, 548
AGS_XML_BUSINESS_GROUP_GET_CLASS, 548
AGS_XML_BUSINESS_GROUP_GET_OBJ_MUTEX, 547
ags_xml_business_group_get_type, 548
ags_xml_business_group_new, 547
ags_xml_business_group_open_filename, 547
AGS_XML_CERTIFICATE, 550
AGS_XML_CERTIFICATE_CLASS, 550
AGS_XML_CERTIFICATE_GET_CLASS, 551
AGS_XML_CERTIFICATE_GET_OBJ_MUTEX, 549
ags_xml_certificate_get_type, 551
ags_xml_certificate_new, 550
ags_xml_certificate_open_filename, 550
AGS_XML_PASSWORD_STORE, 553
AGS_XML_PASSWORD_STORE_CLASS, 554
ags_xml_password_store_find_login, 553
AGS_XML_PASSWORD_STORE_GET_CLASS, 554
AGS_XML_PASSWORD_STORE_GET_OBJ_MUTEX, 552
ags_xml_password_store_get_type, 554
ags_xml_password_store_new, 553
ags_xml_password_store_open_filename, 552
AgsXmlAuthentication, 546
AgsXmlAuthenticationClass, 546
AgsXmlBusinessGroup, 548
AgsXmlBusinessGroupClass, 548
AgsXmlCertificate, 551
AgsXmlCertificateClass, 551
AgsXmlPasswordStore, 554
AgsXmlPasswordStoreClass, 554