lasso Reference Manual

COLLABORATORS

	TITLE : lasso Reference Manual		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 2, 2021	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

List of Figures

Chapter 1

Lasso & Liberty Alliance Overview

Lasso is a library which provides all the necessary functions for sites to implement Liberty Alliance specifications. It defines processes for federated identities, single sign-on and related protocols.

Founded in 2001 by Sun in order to propose an alternative to the Microsoft Passport project, the consortium Liberty Alliance aims to promote an infrastructure of standards allowing the management of federated identities between several services or systems.

A federated identity (or network identity) of an individual or a legal entity on Internet gather at the same time:

- Its identification (name, co-ordinates, preferences, history...);
- Its authentication (which guarantees the validity of an identity);
- Its authorisations (access rights to information, access rights to services).

Liberty standards aims to give more coherence to a network identity which is scattered (numerous logins and passwords) today. This identity becomes frequently delicate to manage, both for customers and businesses.

The Liberty Alliance specifications define three types of actors:

- The user, person or entity who can acquire an identity;
- The identity provider which creates and manages the identity of the users, and authenticates them to the service providers;
- The service provider who provides services to the users once that they have authenticated to an identity provider.

One calls circle of trust a grouping of identity providers and service providers which agreed to share (to federate) the identity of their users.

Contrary to most other implementations of Liberty Alliance, Lasso is not a full-fedged system but a simple C library, with complete bindings for Java, Perl, PHP and Python. The integration work should largely be facilitated. An existing site should be able to integrate it in a few days of development, without calling into question its architecture. Lasso is a library written in C Language.

Lasso is built on top of libxml2, XMLSec and OpenSSL and is licensed under the GNU General Public License (with an OpenSSL exception).

Chapter 2

Application Programming Interface

Chapter 3

Lasso Architecture

Lasso handle the concepts of providers sharing identities, that can enable the creation of sessions following an authentication. The current provider is represented by the LassoServer object, which inherit from the LassoProvider. All known providers to the current provider must be registered inside the LassoServer object. Some providers are identity providers, by sending authentication request you can establish federation, materialised by LassoFederation objects, between identity stored by the identity provider and another stored by the current provider. Those federation are stored in the LassoIdentity object. Each time an authentication is done, an assertion reprenting it is stored in the LassoSession object. They are stored in a map, keyed by the originating identity provider, see lasso_session_get_assertion().

Protocols from the ID-FF 1.2 or the SAML 2.0 family are represented by object whose class inherit from LassoProfile. The constructor of those profile objects needs a LassoServer, and eventually a LassoIdentity and a LassoSession.

3.1 LassoServer

LassoServer — Representation of the current server

Functions

LassoServer *	lasso_server_new ()
LassoServer *	lasso_server_new_from_buffers ()
LassoServer *	lasso_server_new_from_dump ()
lasso_error_t	lasso_server_add_provider ()
lasso_error_t	lasso_server_add_provider2 ()
lasso_error_t	lasso_server_add_provider_from_buffer ()
lasso_error_t	lasso_server_load_metadata ()
void	lasso_server_destroy ()
gchar *	lasso_server_dump ()
LassoProvider *	lasso_server_get_provider ()
lasso_error_t	lasso_server_set_encryption_private_key ()
lasso_error_t	lasso_server_set_encryption_private_key_with_password ()
lasso_error_t	lasso_server_load_affiliation ()
gchar *	lasso_server_get_endpoint_url_by_id ()
GList *	lasso_server_get_filtered_provider_list ()
lasso_error_t	lasso_server_saml2_assertion_setup_signature ()

Types and Values

struct	LassoServer

Description

It holds the data about a provider, other providers it knows, which certificates to use, etc.

Functions

lasso_server_new ()

```
LassoServer~*
lasso_server_new (const gchar *metadata,
                  const gchar *private_key,
                  const gchar *private_key_password,
                  const gchar *certificate);
```

Creates a new LassoServer.

Parameters

metadata	path to the provider metadata file or NULL, for a LECP server	
private_key	path to the the server private key file or NULL.	*[allow-none]*
private_key_password	password to private key if it is encrypted, or NULL.	*[allow-none]*
certificate	path to the server certificate file, or NULL.	*[allow-none]*

Returns

a newly created LassoServer object; or NULL if an error occured

lasso_server_new_from_buffers ()

```
LassoServer~*
lasso_server_new_from_buffers (const gchar *metadata,
                               const gchar *private_key_content,
                               const gchar *private_key_password,
                               const gchar *certificate_content);
```

Creates a new LassoServer.

Parameters

metadata	NULL terminated string containing the content of an ID-FF 1.2 metadata file	
private_key_content	NULL terminated string containing a PEM formatted private key.	*[allow-none]*

private_key_password	a NULL terminated string which is the optional password of the private key.	*[allow-none]*
certificate_content	NULL terminated string containing a PEM formatted X509 certificate.	*[allow-none]*

Returns

a newly created LassoServer object; or NULL if an error occured

lasso_server_new_from_dump ()

```
LassoServer~*
lasso_server_new_from_dump (const gchar *dump);
```

Restores the `dump` to a new LassoServer.

Parameters

dump	XML server dump	

Returns

a newly created LassoServer; or NULL if an error occured

lasso_server_add_provider ()

```
lasso_error_t
lasso_server_add_provider (LassoServer *server,
                           LassoProviderRole role,
                           const gchar *metadata,
                           const gchar *public_key,
                           const gchar *ca_cert_chain);
```

Creates a new LassoProvider and makes it known to the `server`

Parameters

server	a LassoServer	
role	provider role, identity provider or service provider	
metadata	path to the provider metadata file	
public_key	provider public key file (may be a certificate) or NULL.	*[allow-none]*
ca_cert_chain	provider CA certificate chain file or NULL.	*[allow-none]*

Returns

0 on success; a negative value if an error occured.

lasso_server_add_provider2 ()

```
lasso_error_t
lasso_server_add_provider2 (LassoServer *server,
                            LassoProvider *provider);
```

Add *provider* to the list of known providers object of *server* .

Return 0 if successful, LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ otherwise.

Parameters

server	a LassoServer object	
provider	a LassoProvider object	

lasso_server_add_provider_from_buffer ()

```
lasso_error_t
lasso_server_add_provider_from_buffer (LassoServer *server,
                                       LassoProviderRole role,
                                       const gchar *metadata,
                                       const gchar *public_key,
                                       const gchar *ca_cert_chain);
```

Creates a new LassoProvider and makes it known to the *server*

Parameters

server	a LassoServer	
role	provider role, identity provider or service provider	
metadata	a string buffer containg the metadata file for a new provider	
public_key	provider public key file (may be a certificate) or NULL.	*[allow-none]*
ca_cert_chain	provider CA certificate chain file or NULL.	*[allow-none]*

Returns

0 on success; a negative value if an error occured.

lasso_server_load_metadata ()

```
lasso_error_t
lasso_server_load_metadata (LassoServer *server,
                            LassoProviderRole role,
                            const gchar *federation_file,
                            const gchar *trusted_roots,
                            GList *blacklisted_entity_ids,
                            GList **loaded_entity_ids,
                            LassoServerLoadMetadataFlag flags);
```

Load all the SAML 2.0 entities from *federation_file* which contains a declaration for *role* . If *trusted_roots* is non-NULL, use it to check a signature on the metadata file, otherwise ignore signature validation.

Parameters

server	a LassoServer object	
role	a LassoProviderRole value	
federation_file	path to a SAML 2.0 metadata file	
trusted_roots	a PEM encoded files containing the certificates to check signatures on the metadata file (optional).	*[allow-none]*
blacklisted_entity_ids	a list of EntityID which should not be loaded, can be NULL.	*[allow-none][element-type string]*
loaded_entity_ids	an output parameter for the list of the loaded EntityID, can be NULL.	*[transfer full][element-type string][allow-none]*
flags	flags modifying the behaviour for checking signatures on EntityDescriptor and EntitiesDescriptors nodes.	

Returns

0 on success, an error code otherwise, among:

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if server is not a LassoServer object or *role* is not a valid role value,
- LASSO_DS_ERROR_CA_CERT_CHAIN_LOAD_FAILED if the *trusted_root* file cannot be loaded,

lasso_server_destroy ()

```
void
lasso_server_destroy (LassoServer *server);
```

Destroys a server.

Parameters

server	a LassoServer	

lasso_server_dump ()

```
gchar~*
lasso_server_dump (LassoServer *server);
```

Dumps *server* content to an XML string.

Parameters

server	a LassoServer	

Returns

the dump string. It must be freed by the caller.

[transfer full]

lasso_server_get_provider ()

```
LassoProvider~*
lasso_server_get_provider (const LassoServer *server,
                           const gchar *providerID);
```

Looks up for a LassoProvider whose ID is *`providerID`* and returns it.

Parameters

server	a LassoServer	
providerID	the provider ID	

Returns

the LassoProvider, NULL if it was not found. The LassoProvider is owned by Lasso and should not be freed.

[transfer none]

lasso_server_set_encryption_private_key ()

```
lasso_error_t
lasso_server_set_encryption_private_key
                                (LassoServer *server,
                                 const gchar *filename_or_buffer);
```

Warning

`lasso_server_set_encryption_private_key` has been deprecated since version 2.3 and should not be used in newly-written code.

Use lasso_server_set_encryption_private_key_with_password() instead.

Load an encryption private key from a file and set it in the server object

If *`filename_or_buffer`* is NULL, it frees the currently setted key.

Parameters

server	a LassoServer	
filename_or_buffer	file name of the encryption key to load or its content as a NULL-terminated string.	*[allow-none]*

Returns

0 on success; another value if an error occured.

lasso_server_set_encryption_private_key_with_password ()

```
lasso_error_t
lasso_server_set_encryption_private_key_with_password
                               (LassoServer *server,
                                const gchar *filename_or_buffer,
                                const gchar *password);
```

Load an encryption private key from a file and set it in the server object. If `password` is non-NULL try to decrypt the key with it.

If `filename_or_buffer` is NULL, it frees the currently setted key.

Parameters

server	a LassoServer	
filename_or_buffer	file name of the encryption key to load or its content as a NULL-terminated string.	*[allow-none]*
password	an optional password to decrypt the encryption key.	*[allow-none]*

Returns

0 on success; another value if an error occured.

Since: 2.3

lasso_server_load_affiliation ()

```
lasso_error_t
lasso_server_load_affiliation (LassoServer *server,
                               const gchar *filename);
```

Load an affiliation metadata file into `server` ; this must be called after providers have been added to `server` .

Parameters

server	a LassoServer	
filename	file name of the affiliation metadata to load	

Returns

0 on success; another value if an error occured.

lasso_server_get_endpoint_url_by_id ()

```
gchar~*
```

```
lasso_server_get_endpoint_url_by_id (const LassoServer *server,
                                     const gchar *provider_id,
                                     const gchar *endpoint_description);
```

Locate the provider in the server's list of providers, then select an endpoint given the *endpoint_description* and return than endpoint's URL. If the provider cannot be found or if the provider does not have a matching endpoint NULL will be returned.

Parameters

server	a LassoServer	
provider_id	the EntityID whose endpoints will be examined.	
endpoint_description	string describing criteria used to select endpoint.	

Returns

url (must be freed by caller)

lasso_server_get_filtered_provider_list ()

```
GList~*
lasso_server_get_filtered_provider_list
                                (const LassoServer *server,
                                 LassoProviderRole role,
                                 LassoMdProtocolType protocol_type,
                                 LassoHttpMethod http_method);
```

Iterate over the *server* providers and build a list of provider EntityID's who have the specified *role* and at least one endpoint matching the *protocol_type* and *http_method* . Return a GList list of EntityID's at the *provider_list* pointer. The caller is responsible for freeing the *provider_list* by calling lasso_release_list_of_strings().

Parameters

server	a LassoServer	
role	each returned provider will match this LassoProviderRole	
protocol_type	provider must have endpoint matching LassoMdProtocolType and *http_method*	
http_method	provider must have endpoint matching LassoHttpMethod and *protocol_type*	

Returns

GList of matching provider EntityID's returned here.

[transfer full][element-type string]

lasso_server_saml2_assertion_setup_signature ()

```
lasso_error_t
lasso_server_saml2_assertion_setup_signature
                               (LassoServer *server,
                                LassoSaml2Assertion *saml2_assertion);
```

Configure signature on a saml2:Assertion element.

Parameters

server	a LassoServer object	
saml2_assertion	a LassoSaml2Assertion object	

Returns

0 if successfull, an error code otherwise.

Types and Values

struct LassoServer

```
struct LassoServer {
  LassoProvider parent;

  GHashTable *providers; /* of LassoProvider */
  /* Can actually contain LassoDataService or LassoIdWsf2DataService or any subclass */

  gchar *private_key;
  gchar *private_key_password;
  gchar *certificate;
  LassoSignatureMethod signature_method;
};
```

3.2 LassoProvider

LassoProvider — Service or identity provider

Functions

LassoProvider *	lasso_provider_new ()
LassoProvider *	lasso_provider_new_from_buffer ()
gchar *	lasso_provider_get_assertion_consumer_service_url ()
gchar *	lasso_provider_get_metadata_one ()
GList *	lasso_provider_get_metadata_list ()
LassoProvider *	lasso_provider_new_from_dump ()
LassoHttpMethod	lasso_provider_get_first_http_method ()
gboolean	lasso_provider_accept_http_method ()
gboolean	lasso_provider_has_protocol_profile ()
gchar *	lasso_provider_get_base64_succinct_id ()
xmlNode *	lasso_provider_get_organization ()

LassoProtocolConformance	lasso_provider_get_protocol_conformance ()
LassoEncryptionMode	lasso_provider_get_encryption_mode ()
void	lasso_provider_set_encryption_mode ()
void	lasso_provider_set_encryption_sym_key_type ()
lasso_error_t	lasso_provider_verify_single_node_signature ()
gchar *	lasso_provider_get_default_name_id_format ()
const char *	lasso_provider_get_sp_name_qualifier ()
GList *	lasso_provider_get_idp_supported_attributes ()
char *	lasso_provider_get_valid_until ()
char *	lasso_provider_get_cache_duration ()
char *	lasso_provider_get_metadata_one_for_role ()
GList *	lasso_provider_get_metadata_list_for_role ()
GList *	lasso_provider_get_metadata_keys_for_role ()
LassoProviderRole	lasso_provider_get_roles ()
gboolean	lasso_provider_match_conformance ()
void	lasso_provider_set_protocol_conformance ()

Types and Values

struct	LassoProvider
enum	LassoProviderRole
enum	LassoHttpMethod
enum	LassoMdProtocolType
enum	LassoProtocolConformance
enum	LassoEncryptionMode
enum	LassoEncryptionSymKeyType

Description

The LassoProvider object holds metadata about a provider. Metadata are sorted into descriptors, each descriptor being assigned a role. We refer you to *Liberty Metadata Description and Discovery Specification* and *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*.

Roles are represented by the enumeration LassoProviderRole, you can access descriptors content using lasso_provider_get_metadata_list and lasso_provider_get_metadata_by_role(). Descriptors resources are flattened inside a simple hashtable. For example to get the URL(s) for the SAML 2.0 single logout response endpoint using binding HTTP-POST of the SP descriptor of a provider called x, you would call:

```
GList *urls = lasso_provider_get_metadata_list_for_role(x, LASSO_PROVIDER_ROLE_SP, " ↩
    SingleLogoutService HTTP-POST ResponseLocation");
```

A provider usually possess a default role stored in the LassoProvider.role field, which is initialized by the lasso_server_add_provider() method when registering a new remote provider to our current provider. The methods lasso_provider_get_metadata() and lasso_provider_get_metadata_list() use this default role to access descriptors.

Functions

lasso_provider_new ()

```
LassoProvider~*
lasso_provider_new (LassoProviderRole role,
                    const char *metadata,
                    const char *public_key,
                    const char *ca_cert_chain);
```

Creates a new LassoProvider.

Parameters

role	provider role, identity provider or service provider	
metadata	path to the provider metadata file	
public_key	path to the provider public key file (may be a certificate) or NULL.	*[allow-none]*
ca_cert_chain	path to the provider CA certificate chain file or NULL.	*[allow-none]*

Returns

a newly created LassoProvider; or NULL if an error occured

lasso_provider_new_from_buffer ()

```
LassoProvider~*
lasso_provider_new_from_buffer (LassoProviderRole role,
                                const char *metadata,
                                const char *public_key,
                                const char *ca_cert_chain);
```

Creates a new LassoProvider.

Parameters

role	provider role, identity provider or service provider	
metadata	string buffer containing a metadata file	
public_key	path to the provider public key file (may be a certificate) or NULL.	*[allow-none]*
ca_cert_chain	path to the provider CA certificate chain file or NULL.	*[allow-none]*

Returns

a newly created LassoProvider; or NULL if an error occured

lasso_provider_get_assertion_consumer_service_url ()

```
gchar~*
lasso_provider_get_assertion_consumer_service_url
                               (LassoProvider *provider,
                                const char *service_id);
```

Extracts the AssertionConsumerServiceURL from the provider metadata descriptor.

Parameters

provider	a LassoProvider	
service_id	the AssertionConsumerServiceID, NULL for default.	*[allow-none]*

Returns

the element value, NULL if the element was not found. This string must be freed by the caller.

[allow-none][transfer full]

lasso_provider_get_metadata_one ()

```
gchar~*
lasso_provider_get_metadata_one (LassoProvider *provider,
                                 const char *name);
```

Extracts the element `name` from the provider metadata descriptor.

Parameters

provider	a LassoProvider	
name	the element name	

Returns

the element value, NULL if the element was not found. This string must be freed by the caller.

[transfer full][allow-none]

lasso_provider_get_metadata_list ()

```
GList~*
lasso_provider_get_metadata_list (LassoProvider *provider,
                                  const char *name);
```

Extracts zero to many elements from the provider metadata descriptor.

Parameters

provider	a LassoProvider	
name	the element name	

Returns

a GList with the elements. This GList is internally allocated and points to internally allocated strings. It must not be freed, modified or stored.

[transfer none][element-type string]

lasso_provider_new_from_dump ()

```
LassoProvider~*
lasso_provider_new_from_dump (const gchar *dump);
```

Restores the *dump* to a new LassoProvider.

Parameters

dump	XML provider dump	

Returns

a newly created LassoProvider; or NULL if an error occured.

lasso_provider_get_first_http_method ()

```
LassoHttpMethod
lasso_provider_get_first_http_method (LassoProvider *provider,
                                      LassoProvider *remote_provider,
                                      LassoMdProtocolType protocol_type);
```

Looks up and returns a LassoHttpMethod appropriate for performing the *protocol_type* between *provider* and *remote_provider* .

Parameters

provider	a LassoProvider.	*[transfer none]*
remote_provider	a LassoProvider depicting the remote provider	
protocol_type	a Liberty profile	

Returns

the LassoHttpMethod

lasso_provider_accept_http_method ()

```
gboolean
lasso_provider_accept_http_method (LassoProvider *provider,
                                   LassoProvider *remote_provider,
                                   LassoMdProtocolType protocol_type,
                                   LassoHttpMethod http_method,
                                   gboolean initiate_profile);
```

Gets if *http_method* is an appropriate method for the *protocol_type* profile between *provider* and *remote_provider* .

Parameters

provider	a LassoProvider	
remote_provider	a LassoProvider depicting the remote provider	
protocol_type	a Liberty profile type	

http_method	an HTTP method	
initiate_profile	whether *provider* initiates the profile	

Returns

TRUE if it is appropriate

lasso_provider_has_protocol_profile ()

```
gboolean
lasso_provider_has_protocol_profile (LassoProvider *provider,
                                     LassoMdProtocolType protocol_type,
                                     const char *protocol_profile);
```

Gets if *provider* supports *protocol_profile* .

Parameters

provider	a LassoProvider	
protocol_type	a Liberty profile type	
protocol_profile	a fully-qualified Liberty profile	

Returns

TRUE if it is supported

lasso_provider_get_base64_succinct_id ()

```
gchar~*
lasso_provider_get_base64_succinct_id (const LassoProvider *provider);
```

Computes and returns the base64-encoded provider succinct ID.

Parameters

provider	a LassoProvider	

Returns

the provider succinct ID. This string must be freed by the caller.

[transfer full][allow-none]

lasso_provider_get_organization ()

```
xmlNode~*
lasso_provider_get_organization (const LassoProvider *provider);
```

Returns the provider metadata <Organization> XML node.

Parameters

provider	a LassoProvider

Returns

the <Organization/> node (libxml2 xmlNode*); or NULL if it is not found. This xmlnode must be freed by the caller.

[transfer full][allow-none]

lasso_provider_get_protocol_conformance ()

```
LassoProtocolConformance
lasso_provider_get_protocol_conformance
                                 (const LassoProvider *provider);
```

Return the protocol conformance of the given provider, it should allow to switch behaviour of SP and IdP code toward a specific protocol. See also LassoProtocolConformance.

Parameters

provider	a LassoProvider object

Returns

a value in the LassoProtocolConformance enumeration.

lasso_provider_get_encryption_mode ()

```
LassoEncryptionMode
lasso_provider_get_encryption_mode (LassoProvider *provider);
```

Return the current encryption mode.

Parameters

provider	a LassoProvider object

Returns

a value in the LassoEncryptionMode enumeration.

lasso_provider_set_encryption_mode ()

```
void
lasso_provider_set_encryption_mode (LassoProvider *provider,
                                     LassoEncryptionMode encryption_mode);
```

Activate or desactivate encryption

Parameters

provider	provider to set encryption for	
encryption_mode	TRUE to activate, FALSE to desactivate	

lasso_provider_set_encryption_sym_key_type ()

```
void
lasso_provider_set_encryption_sym_key_type
                               (LassoProvider *provider,
                                LassoEncryptionSymKeyType encryption_sym_key_type);
```

Set the type of the generated encryption symetric key

Parameters

provider	provider to set encryption for	
encryption_sym_key_type	enum type for generated symetric key	

lasso_provider_verify_single_node_signature ()

```
lasso_error_t
lasso_provider_verify_single_node_signature
                               (LassoProvider *provider,
                                LassoNode *node,
                                const char *id_attr_name);
```

Return wheter the provider signed this node.

Parameters

provider	a LassoProvider object	
node	a LassoNode object, still having its originalXmlnode content, and containing an XML signature.	
id_attr_name	the name of the ID attribute to lookup.	

Returns

0 if the node is signed by this provider, an error code otherwise.

lasso_provider_get_default_name_id_format ()

```
gchar~*
lasso_provider_get_default_name_id_format
                               (LassoProvider *provider);
```

If the provider has a list of supported name id formats in its metadatas, return the first one.

Parameters

provider	a LassoProvider object

Returns

a NameIDFormat URI or NULL, the returned value must be freed by the caller.

[transfer full][allow-none]

lasso_provider_get_sp_name_qualifier ()

```
const char~*
lasso_provider_get_sp_name_qualifier (LassoProvider *provider);
```

Return the entityID to use for qualifying NameIdentifier.

Parameters

provider	a LassoPRovider object

Returns

a private string or NULL. Do not keep a reference on this string or free it.

[transfer none][allow-none]

lasso_provider_get_idp_supported_attributes ()

```
GList~*
lasso_provider_get_idp_supported_attributes
                               (LassoProvider *provider);
```

If the provider supports the IDP SSO role, then return the list of Attribute definition that this provider declared supporting.

Parameters

provider	a LassoProvider object

Returns

a list of LassoSaml2Attribute or LassoSamlAttribute.

[transfer none][element-type LassoNode]

lasso_provider_get_valid_until ()

```
char~*
lasso_provider_get_valid_until (LassoProvider *provider);
```

Return the time after which the metadata for this provider will become invalid. This is an ISO-8601 formatted string.

Parameters

provider	a LassoProvider object

Returns

an internally allocated string, you can copy it but not store it.

[transfer none]

lasso_provider_get_cache_duration ()

```
char~*
lasso_provider_get_cache_duration (LassoProvider *provider);
```

Return the time during which the metadata for this provider can be kept.

Parameters

provider	a LassoProvider object

Returns

an internally allocated string, you can copy it but not store it.

[transfer none]

lasso_provider_get_metadata_one_for_role ()

```
char~*
lasso_provider_get_metadata_one_for_role
                                (LassoProvider *provider,
                                 LassoProviderRole role,
                                 const char *name);
```

Return the given information extracted from the metadata of the given LassoProvider for the given `role` descriptor.

Retun value: a newly allocated string or NULL. If non-NULL must be freed by the caller.

Parameters

provider	a LassoProvider object
role	a LassoProviderRole value
name	a metadata information name

lasso_provider_get_metadata_list_for_role ()

```
GList~*
lasso_provider_get_metadata_list_for_role
                                (const LassoProvider *provider,
                                 LassoProviderRole role,
                                 const char *name);
```

Extracts zero to many elements from the `provider` descriptor for the given `role` .

Parameters

provider	a LassoProvider	
role	a LassoProviderRole value	
name	the element name	

Returns

a GList with the elements. This GList is internally allocated and points to internally allocated strings. It must not be freed, modified or stored.

[transfer none][element-type string]

lasso_provider_get_metadata_keys_for_role ()

```
GList~*
lasso_provider_get_metadata_keys_for_role
                                (LassoProvider *provider,
                                 LassoProviderRole role);
```

Returns the list of metadata keys existing for the given provider.

Parameters

provider	a LassoProvider object	
role	a LassoProviderRole value	

Returns

a newly allocated list of strings.

[element-type utf8][transfer full]

lasso_provider_get_roles ()

```
LassoProviderRole
lasso_provider_get_roles (LassoProvider *provider);
```

Return the bitmask of the supported roles.

Parameters

provider	a LassoProvider object	

Returns

a LassoProviderRole enumeration value.

lasso_provider_match_conformance ()

```
gboolean
lasso_provider_match_conformance (LassoProvider *provider,
                                  LassoProvider *another_provider);
```

Return whether the two provider support a same protocol. See also LassoProtocolConformance.

Parameters

provider	a LassoProvider object
another_provider	a LassoProvider object

Returns

TRUE or FALSE.

lasso_provider_set_protocol_conformance ()

```
void
lasso_provider_set_protocol_conformance
                               (LassoProvider *provider,
                                LassoProtocolConformance protocol_conformance);
```

Normally the protocol conformance is set when the metadata for the provider is loaded because the metadata defines the type of server. However some LassoServer variants do not have metadata (e.g. ECP) therefore instead of loading the metadata it is necessary to explicitly set the protocol conformance because parts of the Lasso library dispatch based on the protocol conformance. Without the protocol conformance being set it is likely the wrong code will execute.

WARNING, do not manually set the protocol conformance if metadata has been loaded, metadata is the final arbiter of protocol conformance.

Parameters

provider	a LassoProvider object
protocol_conformance	LassoProtocolConformance enumerated value.

Returns

0 on success; another value if an error occured.

Types and Values

struct LassoProvider

```
struct LassoProvider {
  LassoNode parent;

  gchar *ProviderID;
  LassoProviderRole role;

  char *metadata_filename;
  gchar *public_key;
  gchar *ca_cert_chain;
};
```

Any kind of provider, identity provider, service provider, attribute authority, authorization authority will be represented by a LassoProvider object. This object will holds public keys, certificate chains and metadata informations. The ID-FF 1.2 and SAML 2.0 metadata files are flattened inside a key-value map that you can access using the functions lasso_provider_get_metadata_one_for_role(), lasso_provider_get_metadata_list_for_role(), lasso_provider_get_metadata_keys_for_role().

Members

LassoNode *parent*;	
gchar **ProviderID*;	the identifier URI of this provider
LassoProviderRole *role*;	the role prescribed when this LassoProvider was built
char **metadata_filename*;	file path or content of the metadata description for this provider.
gchar **public_key*;	file path or content of the public key file for this provider.

gchar **ca_cert_chain*;	file path or content of the CA cert chain used to validate signature of this provider (can be used instead of a public key to limit the need for metadata updates).

enum LassoProviderRole

LassoProviderRole is an enumeration allowing to enumerate the roles handled by a provider, it can be used in a bitmask as each value is a power of 2 (except LASSO_PROVIDER_ROLE_ANY which is the full bitmask and LASSO_PROVIDER_ROLE_NONE).

Members

LASSO_PROVIDER_ROLE_ANY	
LASSO_PROVIDER_ROLE_NONE	unitialized value (internal use)

LASSO_PROVIDER_ROLE_SP	service provider.
LASSO_PROVIDER_ROLE_IDP	identity provider.
LASSO_PROVIDER_ROLE_BOTH	service&identity provider.
LASSO_PROVIDER_ROLE_AUTHN_AUTHORITY	an authentification authority, i.e. an endpoint able to return previously returned assertion,

LASSO_PROVIDER_ROLE_AUTHZ_AUTHORITY	an authorization authority, i.e. an endpoint able to return assertion providing authorization about a principal acessing a resource,
LASSO_PROVIDER_ROLE_ATTRIBUTE_AUTHORITY	an attribute authority, i.e. an endpoint able to return attributes aboute a principal,

LASSO_PROVIDER_ROLE_LAST	all values in the enumeration are guaranteed to be < to *LASSO_PROVIDER_ROLE_LAST* .
LASSO_PROVIDER_ROLE_ALL	

enum LassoHttpMethod

Method.

Members

LASSO_HTTP_METHOD_NONE	invalid value (internal use)
LASSO_HTTP_METHOD_ANY	any method will do
LASSO_HTTP_METHOD_IDP_INITIATED	not a method, for IdP initiated profile
LASSO_HTTP_METHOD_GET	HTTP GET
LASSO_HTTP_METHOD_POST	Browser POST
LASSO_HTTP_METHOD_REDIRECT	HTTP-Redirect based

LASSO_HTTP_METHOD_SOAP	SOAP/HTTP based
LASSO_HTTP_METHOD_ARTIFACT_GET	Artifact by HTTP GET (SAML 2.0)
LASSO_HTTP_METHOD_ARTIFACT_POST	Artifact by HTTP POST (SAML 2.0)
LASSO_HTTP_METHOD_PAOS	PAOS/HTTP based (SAML 2.0)
LASSO_HTTP_METHOD_LAST	

enum LassoMdProtocolType

Liberty Metadata Type.

Members

LASSO_MD_PROTOCOL_TYPE_FEDERATION_TERMINATION	Federation Termination Notification
LASSO_MD_PROTOCOL_TYPE_NAME_IDENTIFIER_MAPPING	Name Identifier Mapping
LASSO_MD_PROTOCOL_TYPE_REGISTER_NAME_IDENTIFIER	Name Registration
LASSO_MD_PROTOCOL_TYPE_SINGLE_LOGOUT	Single Logout

LASSO_MD_PROTOCOL_TYPE_SINGLE_SIGN_ON	Single Sign-On and Federation
LASSO_MD_PROTOCOL_TYPE_ARTIFACT_RESOLUTION	Artifact Resolution (SAML 2.0)
LASSO_MD_PROTOCOL_TYPE_MANAGE_NAME_ID	Manage Name Identifier (SAML 2.0)
LASSO_MD_PROTOCOL_TYPE_ASSERTION_ID_REQUEST	Assertion ID Request (SAML 2.0)
LASSO_MD_PROTOCOL_TYPE_AUTHN_QUERY	
LASSO_MD_PROTOCOL_TYPE_AUTHZ	
LASSO_MD_PROTOCOL_TYPE_ATTRIBUTE	
LASSO_MD_PROTOCOL_TYPE_LAST	

enum LassoProtocolConformance

Provider protocol conformance.

Members

LASSO_PROTOCOL_NONE	
LASSO_PROTOCOL_LIBERTY_1_0	Liberty ID-FF 1.0
LASSO_PROTOCOL_LIBERTY_1_1	Liberty ID-FF 1.1
LASSO_PROTOCOL_LIBERTY_1_2	Liberty ID-FF 1.2 / ID-WSF 1.0

LASSO_PROTOCOL_SAML_2_0	SAML 2.0

enum LassoEncryptionMode

Encryption mode.

Members

LASSO_ENCRYPTION_MODE_NONE	Encrypt nothing
LASSO_ENCRYPTION_MODE_NAMEID	Encrypt NameIDs
LASSO_ENCRYPTION_MODE_ASSERTION	Encrypt Assertions

enum LassoEncryptionSymKeyType

Encryption symetric key type.

Members

LASSO_ENCRYPTION_SYM_KEY_TYPE_DEFAULT	Default type (AES 128)
LASSO_ENCRYPTION_SYM_KEY_TYPE_AES_256	Aes 256 bits key
LASSO_ENCRYPTION_SYM_KEY_TYPE_AES_128	Aes 128 bits key
LASSO_ENCRYPTION_SYM_KEY_TYPE_3DES	Triple DES 192 bits key
LASSO_ENCRYTPION_SYM_KEY_TYPE_LAST	

3.3 LassoIdentity

LassoIdentity — Principal identity

Functions

LassoIdentity *	lasso_identity_new ()
LassoIdentity *	lasso_identity_new_from_dump ()
LassoFederation *	lasso_identity_get_federation ()
void	lasso_identity_destroy ()
gchar *	lasso_identity_dump ()

Types and Values

struct	LassoIdentity

Description

A LassoIdentity object records the identifers that a principal use two federate pairs of providers.

Functions

lasso_identity_new ()

```
LassoIdentity~*
lasso_identity_new (void);
```

Creates a new LassoIdentity.

Returns

a newly created LassoIdentity

lasso_identity_new_from_dump ()

```
LassoIdentity~*
lasso_identity_new_from_dump (const gchar *dump);
```

Restores the *`dump`* to a new LassoIdentity.

Parameters

dump	XML server dump	

Returns

a newly created LassoIdentity; or NULL if an error occured

lasso_identity_get_federation ()

```
LassoFederation~*
lasso_identity_get_federation (LassoIdentity *identity,
                               const char *providerID);
```

Looks up and returns the LassoFederation for this provider ID.

Parameters

identity	a LassoIdentity
providerID	the provider ID

Returns

the LassoFederation; or NULL if it didn't exist. The LassoFederation is internally allocated. It must not be freed, modified or stored.

[transfer none]

lasso_identity_destroy ()

```
void
lasso_identity_destroy (LassoIdentity *identity);
```

Destroys an identity.

Parameters

identity	a LassoIdentity

lasso_identity_dump ()

```
gchar~*
lasso_identity_dump (LassoIdentity *identity);
```

Dumps `identity` content to an XML string.

Parameters

identity	a LassoIdentity

Returns

the dump string. It must be freed by the caller.

[transfer full]

Types and Values

struct LassoIdentity

```
struct LassoIdentity {
  LassoNode parent;

  GHashTable *federations; /* of LassoFederation */
  gboolean is_dirty;
};
```

3.4 LassoFederation

LassoFederation — Principal federation between two providers

Functions

LassoFederation *	lasso_federation_new ()
void	lasso_federation_build_local_name_identifier ()
void	lasso_federation_destroy ()
gboolean	lasso_federation_verify_name_identifier ()

Types and Values

struct	LassoFederation

Description

A LassoFederation represents the an identifier shared by two provider, usually an identity provider and a service provider. Instance of this class are usually never accessed directly.

Functions

lasso_federation_new ()

```
LassoFederation~*
lasso_federation_new (const gchar *remote_providerID);
```

Creates a new LassoFederation with the remote provider.

Parameters

remote_providerID	remote Provider ID

Returns

a newly created LassoFederation

lasso_federation_build_local_name_identifier ()

```
void
lasso_federation_build_local_name_identifier
                                (LassoFederation *federation,
                                 const gchar *nameQualifier,
                                 const gchar *format,
                                 const gchar *content);
```

Builds federation local name identifier.

Parameters

federation	a LassoFederation	
nameQualifier	the name identifier qualifier	
format	the name identifier format	
content	the name identifier content	

lasso_federation_destroy ()

```
void
lasso_federation_destroy (LassoFederation *federation);
```

Destroys a federation.

Parameters

federation	a LassoFederation	

lasso_federation_verify_name_identifier ()

```
gboolean
lasso_federation_verify_name_identifier
                               (LassoFederation *federation,
                                LassoNode *name_identifier);
```

Checks whether federation is for *name_identifier* .

Parameters

federation	a LassoFederation	
name_identifier	the LassoSamlNameIdentifier	

Returns

TRUE if the federation is for *name_identifier* .

Types and Values

struct LassoFederation

```
struct LassoFederation {
  LassoNode parent;

  gchar *remote_providerID;
  LassoNode *local_nameIdentifier;
  LassoNode *remote_nameIdentifier;
};
```

3.5 LassoSession

LassoSession — Principal Session

Functions

LassoSession *	lasso_session_new ()
LassoSession *	lasso_session_new_from_dump ()
gchar *	lasso_session_dump ()
void	lasso_session_destroy ()
GList *	lasso_session_get_assertions ()
LassoNode *	lasso_session_get_assertion ()
lasso_error_t	lasso_session_remove_assertion ()
lasso_error_t	lasso_session_add_assertion ()
gchar *	lasso_session_get_provider_index ()
gboolean	lasso_session_is_empty ()

Types and Values

struct	LassoSession

Description

Functions

lasso_session_new ()

```
LassoSession~*
lasso_session_new (void);
```

Creates a new LassoSession.

Returns

a newly created LassoSession

lasso_session_new_from_dump ()

```
LassoSession~*
lasso_session_new_from_dump (const gchar *dump);
```

Restores the *dump* to a new LassoSession.

Parameters

dump	XML server dump	

Returns

a newly created LassoSession; or NULL if an error occured

lasso_session_dump ()

```
gchar~*
lasso_session_dump (LassoSession *session);
```

Dumps *session* content to an XML string.

Parameters

session	a LassoSession

Returns

the dump string. It must be freed by the caller.

[transfer full]

lasso_session_destroy ()

```
void
lasso_session_destroy (LassoSession *session);
```

Destroys a session.

Parameters

session	a LassoSession

lasso_session_get_assertions ()

```
GList~*
lasso_session_get_assertions (LassoSession *session,
                              const char *provider_id);
```

Gets the assertions for the given *provider_id* .

Parameters

session	a LassoSession
provider_id	the provider ID

Returns

a list of LassoSamlAssertion.

[allow-none][transfer container][element-type LassoNode]

lasso_session_get_assertion ()

```
LassoNode~*
lasso_session_get_assertion (LassoSession *session,
                             const gchar *providerID);
```

Gets the assertion for the given *providerID* .

Parameters

session	a LassoSession	
providerID	the provider ID	

Returns

the assertion or NULL if it didn't exist. This LassoSamlAssertion is internally allocated and must not be freed by the caller.

[transfer none][allow-none]

lasso_session_remove_assertion ()

```
lasso_error_t
lasso_session_remove_assertion (LassoSession *session,
                                const gchar *providerID);
```

Removes assertion for `providerID` from `session` .

Parameters

session	a LassoSession	
providerID	the provider ID	

Returns

0 on success; or a negative value otherwise.

lasso_session_add_assertion ()

```
lasso_error_t
lasso_session_add_assertion (LassoSession *session,
                             const char *providerID,
                             LassoNode *assertion);
```

Adds `assertion` to the principal session. This function also add the assertion to the index by assertionID.

Parameters

session	a LassoSession	
providerID	the provider ID	
assertion	the assertion	

Returns

0 on success; or a negative value otherwise.

lasso_session_get_provider_index ()

```
gchar~*
lasso_session_get_provider_index (LassoSession *session,
                                  gint index);
```

Looks up and returns the nth provider id.

Parameters

session	a LassoSession
index	index of requested provider

Returns

the provider id; or NULL if there were no nth provider. This string must be freed by the caller.

[transfer full][allow-none]

lasso_session_is_empty ()

```
gboolean
lasso_session_is_empty (LassoSession *session);
```

Returns TRUE if session is empty.

Parameters

session	a LassoSession

Returns

TRUE if empty

Types and Values

struct LassoSession

```
struct LassoSession {
  LassoNode parent;

  /* Can actually contain LassoSamlAssertion or LassoSaml2Assertion */
  GHashTable *assertions; /* of LassoNode */
  gboolean is_dirty;
};
```

LassoSession stores the assertions received or emitted during the current session. It stores state for using profiles like LassoLogin or LassoLogout.

Members

LassoNode *parent*;		

GHashTable *`assertions`;	a hashtable of LassoSamlAssertion or *[element-type string LassoNode]* LassoSaml2Assertion, indexed by provider ids,.
gboolean `is_dirty`;	whether this session object has been modified since its creation.

3.6 LassoProfile

LassoProfile — Base class for all identity profiles

Functions

LassoRequestType	lasso_profile_get_request_type_from_soap_msg ()
lasso_error_t	lasso_profile_set_soap_fault_response ()
gboolean	lasso_profile_is_liberty_query ()
LassoIdentity *	lasso_profile_get_identity ()
LassoSession *	lasso_profile_get_session ()
gboolean	lasso_profile_is_identity_dirty ()
gboolean	lasso_profile_is_session_dirty ()
lasso_error_t	lasso_profile_set_identity_from_dump ()
lasso_error_t	lasso_profile_set_session_from_dump ()
LassoNode *	lasso_profile_get_nameIdentifier ()
char *	lasso_profile_get_artifact ()
char *	lasso_profile_get_artifact_message ()
void	lasso_profile_set_artifact_message ()
LassoServer *	lasso_profile_get_server ()
void	lasso_profile_set_signature_hint ()
LassoProfileSignatureHint	lasso_profile_get_signature_hint ()
void	lasso_profile_set_signature_verify_hint ()

LassoProfileSignatureVerifyHint	lasso_profile_get_signature_verify_hint ()
LassoProviderRole	lasso_profile_sso_role_with ()
lasso_error_t	lasso_profile_get_signature_status ()

Types and Values

struct	LassoProfile
enum	LassoRequestType
enum	LassoProfileSignatureHint
enum	LassoProfileSignatureVerifyHint

Description

Functions

lasso_profile_get_request_type_from_soap_msg ()

```
LassoRequestType
lasso_profile_get_request_type_from_soap_msg
                                (const gchar *soap);
```

Looks up and return the type of the request in a SOAP message.

Parameters

soap	the SOAP message	

Returns

the type of request

lasso_profile_set_soap_fault_response ()

```
lasso_error_t
lasso_profile_set_soap_fault_response (LassoProfile *profile,
                                       const char *faultcode,
                                       const char *faultstring,
                                       GList *details);
```

Set the response to a SOAP fault, using `faultcode` , `faultstring` , and `details` to initialize it.

Parameters

profile	a LassoProfile object	
faultcode	the code for the SOAP fault	
faultstring	the description for the SOAP fault.	*[allow-none]*
details	a list of nodes to add as details.	*[element-type LassoNode][allow-none]*

Returns

0 if successful, an error code otherwise.

lasso_profile_is_liberty_query ()

```
gboolean
lasso_profile_is_liberty_query (const gchar *query);
```

Tests the query string to know if the URL is called as the result of a Liberty redirect (action initiated elsewhere) or not.

Parameters

query	HTTP query string

Returns

TRUE if Liberty query, FALSE otherwise

lasso_profile_get_identity ()

```
LassoIdentity~*
lasso_profile_get_identity (LassoProfile *profile);
```

Gets the identity bound to `profile` .

Parameters

profile	a LassoProfile

Returns

the identity or NULL if it none was found. The LassoIdentity object is internally allocated and must not be freed by the caller. *[transfer none]*

lasso_profile_get_session ()

```
LassoSession~*
lasso_profile_get_session (LassoProfile *profile);
```

Gets the session bound to `profile` .

Parameters

profile	a LassoProfile

Returns

the session or NULL if it none was found. The LassoSession object is internally allocated and must not be freed by the caller. *[transfer none]*

lasso_profile_is_identity_dirty ()

```
gboolean
lasso_profile_is_identity_dirty (LassoProfile *profile);
```

Checks whether identity has been modified (and should therefore be saved).

Parameters

profile	a LassoProfile

Returns

TRUE if identity has changed

lasso_profile_is_session_dirty ()

```
gboolean
lasso_profile_is_session_dirty (LassoProfile *profile);
```

Checks whether session has been modified (and should therefore be saved).

Parameters

profile	a LassoProfile

Returns

TRUE if session has changed

lasso_profile_set_identity_from_dump ()

```
lasso_error_t
lasso_profile_set_identity_from_dump (LassoProfile *profile,
                                      const gchar *dump);
```

Builds a new LassoIdentity object from XML dump and binds it to *profile* .

Parameters

profile	a LassoProfile
dump	XML identity dump

Returns

0 on success; or a negative value otherwise.

lasso_profile_set_session_from_dump ()

```
lasso_error_t
```

```
lasso_profile_set_session_from_dump (LassoProfile *profile,
                                     const gchar *dump);
```

Builds a new LassoSession object from XML dump and binds it to *profile* .

Parameters

profile	a LassoProfile	
dump	XML session dump	

Returns

0 on success; or a negative value otherwise.

lasso_profile_get_nameIdentifier ()

```
LassoNode~*
lasso_profile_get_nameIdentifier (LassoProfile *profile);
```

Looks up appropriate federation in object and gets the service provider name identifier (which is actually a LassoSamlNameIdentifier in ID-FF 1.2 and LassoSaml2NameID in SAML 2.0).

Parameters

profile	a LassoProfile	

Returns

the name identifier or NULL if none was found. The LassoNode object is internally allocated and must not be freed by the caller.

[transfer none]

lasso_profile_get_artifact ()

```
char~*
lasso_profile_get_artifact (LassoProfile *profile);
```

Return the artifact token

Parameters

profile	a LassoProfile object	

Returns

a newly allocated string or NULL.

[transfer full][allow-none]

lasso_profile_get_artifact_message ()

```
char~*
lasso_profile_get_artifact_message (LassoProfile *profile);
```

Return the artifact message.

Parameters

profile	a LassoProfile object

Returns

a newly allocated string or NULL.

[transfer full][allow-none]

lasso_profile_set_artifact_message ()

```
void
lasso_profile_set_artifact_message (LassoProfile *profile,
                                    const char *message);
```

Set `message` as the content for the ArtifactResolve response.

Parameters

profile	a LassoProfile object
message	the artifact message content

lasso_profile_get_server ()

```
LassoServer~*
lasso_profile_get_server (LassoProfile *profile);
```

Return the LassoServer linked to this profile object. A profile object should always contains one. It allows to find metadatas of other providers and to know our own metadatas.

Parameters

profile	a LassoProfile object

Returns

a LassoServer or NULL if profile is not a LassoProfile or no LassoServer object was setup at the creation of this profile.

[transfer none]

lasso_profile_set_signature_hint ()

```
void
lasso_profile_set_signature_hint (LassoProfile *profile,
                                  LassoProfileSignatureHint signature_hint);
```

By default each profile will choose to sign or not its messages, this method allow to force or forbid the signature of messages, on a per transaction basis.

Parameters

profile	a LassoProfile object	
signature_hint	wheter next produced messages should be signed or not (or let Lasso choose from implicit information).	

lasso_profile_get_signature_hint ()

```
LassoProfileSignatureHint
lasso_profile_get_signature_hint (LassoProfile *profile);
```

Return the value of the signature hint attribute (see lasso_profile_set_signature_hint()).

Parameters

profile	a LassoProfile object	

Returns

a value in the enum type LassoProfileSignatureHint.

lasso_profile_set_signature_verify_hint ()

```
void
lasso_profile_set_signature_verify_hint
                               (LassoProfile *profile,
                                LassoProfileSignatureVerifyHint signature_verify_hint);
```

By default each profile will choose to verify or not its messages, this method allow to force or forbid the signature of messages, on a per transaction basis.

Parameters

profile	a LassoProfile object	
signature_verify_hint	whether next received message signatures should be checked or not (or let Lasso choose from implicit information).	

lasso_profile_get_signature_verify_hint ()

```
LassoProfileSignatureVerifyHint
lasso_profile_get_signature_verify_hint
                               (LassoProfile *profile);
```

Return the value of the signature verify hint attribute (see lasso_profile_set_signature_verify_hint()).

Parameters

profile	a LassoProfile object

Returns

a value in the enum type LassoProfileSignatureVerifyHint.

lasso_profile_sso_role_with ()

```
LassoProviderRole
lasso_profile_sso_role_with (LassoProfile *profile,
                             const char *remote_provider_id);
```

Returns whether the current provider is a service provider relatively to another provider. It uses the LassoProfile.identity to find if a federation qualifier by the given provider exists or the reverse.

Parameters

profile	a LassoProfile object
remote_provider_id	the identifier of a provider

Returns

LASSO_PROVIDER_ROLE_NONE if nothing can be said, LASSO_PROVIDER_ROLE_SP if a federation qualifier by *remote_provi* exists or LASSO_PROVIDER_ROLE_IDP if a federation qualifier by our own LassoProvider.ProviderID exists.

lasso_profile_get_signature_status ()

```
lasso_error_t
lasso_profile_get_signature_status (LassoProfile *profile);
```

Returns the signature status from the last parsed message.

Parameters

profile	a LassoProfile object

Returns

0 if no error from signature checking occurred, an error code otherwise.

Types and Values

struct LassoProfile

```
struct LassoProfile {
  LassoNode parent;

  LassoServer *server;
  LassoNode *request;
  LassoNode *response;

  LassoNode *nameIdentifier;

  gchar *remote_providerID;

  gchar *msg_url;
  gchar *msg_body;
  gchar *msg_relayState;
};
```

LassoProfile, child class of LassoNode is the basis object of profiles object like LassoLogin, LassoLogout, LassoDefederation, LassoNameIdentifierMapping, LassoNameRegistration, LassoNameIdManagement or LassoAssertionQuery. It handles the minimal state used by all theses profiles.

Members

LassoNode *parent*;		
LassoServer **server*;		LassoServer object representing the provider initiating this profile,
LassoNode **request*;		the currently initialized request, or the last request parsed,

LassoNode *response;	the currently initialized request, or the last response parsed,
LassoNode *nameIdentifier;	for profiles which transmit a name identifier (that is, most of them), the parsed name identifier, can be a LassoSamlNameIdentifier or a LassoSaml2NameID,

gchar **remote_providerID*;	the provider ID of the issuer of the last parsed message, whatever it is (a request or a response),
gchar **msg_url*;	when generating a request or a response, it give the URL to contact

gchar *msg_body;	when generating a request or a response using HTTP POST binding (can be HTTP-SOAP or HTTP-Post binding), the body of the POST will be in this field,
gchar *msg_relayState;	put there the relaystate to put in the genereated URL for HTTP-Redirect or HTTP-Get binding.

enum LassoRequestType

Request types (known for SOAP endpoints)

Members

LASSO_REQUEST_TYPE_INVALID	invalid
LASSO_REQUEST_TYPE_LOGIN	Single Sign On and Federation
LASSO_REQUEST_TYPE_LOGOUT	Single Logout
LASSO_REQUEST_TYPE_DEFEDERATION	Federation Termination
LASSO_REQUEST_TYPE_NAME_REGISTRATION	Name Registration
LASSO_REQUEST_TYPE_NAME_IDENTIFIER_MAPPING	Name Identifier Mapping
LASSO_REQUEST_TYPE_LECP	Liberty-Enabled Client / Proxy
LASSO_REQUEST_TYPE_DISCO_QUERY	ID-WSF 1.0 Discovery Query request

LASSO_REQUEST_TYPE_DISCO_MODIFY	ID-WSF 1.0 Discovery Modify Request
LASSO_REQUEST_TYPE_DST_QUERY	ID-WSF 1.0 Data Service Template Query request
LASSO_REQUEST_TYPE_DST_MODIFY	ID-WSF 1.0 Data Service Temaplte Modify request
LASSO_REQUEST_TYPE_SASL_REQUEST	ID-WSF 1.0 Authentication request
LASSO_REQUEST_TYPE_NAME_ID_MANAGEMENT	SAML 2.0 NameID Management request

LASSO_REQUEST_TYPE_IDWSF2_DISCO_SVCMD_REGISTER	ID-WSF 2.0 Discovery Service Metadata Register request
LASSO_REQUEST_TYPE_IDWSF2_DISCO_SVCMD_ASSOCIATION_ADD	ID-WSF 2.0 Discovery Service Metadata Add Association request
LASSO_REQUEST_TYPE_IDWSF2_DISCO_QUERY	ID-WSF 2.0 Discovery Query request

enum LassoProfileSignatureHint

Advice a LassoProfile object about the policy for generating request and response signatures.

Members

LASSO_PROFILE_SIGNATURE_HINT_MAYBE	let Lasso decide what to do.
LASSO_PROFILE_SIGNATURE_HINT_FORCE	generate and validate all signatures.
LASSO_PROFILE_SIGNATURE_HINT_FORBID	do not generate or validate any signature.

enum LassoProfileSignatureVerifyHint

Advice a LassoProfile object about the policy checking request and response signatures.

Members

LASSO_PROFILE_SIGNATURE_VERIFY_HINT_MAYBE	let Lasso decide what to do.
LASSO_PROFILE_SIGNATURE_VERIFY_HINT_FORCE	always check signatures.

LASSO_PROFILE_SIGNATURE_VERIFY_HINT_IGNORE	check signatures but do not stop protocol handling on failures. The result of signature checking is still available in LassoProfile.signature_status
LASSO_PROFILE_SIGNATURE_VERIFY_HINT_LAST	

3.7 Error Codes

Error Codes — Error codes returned by lasso functions

Types and Values

#define	LASSO_ERROR_UNDEFINED
#define	LASSO_ERROR_UNIMPLEMENTED
#define	LASSO_ERROR_OUT_OF_MEMORY
#define	LASSO_ERROR_CAST_FAILED
#define	LASSO_XML_ERROR_NODE_NOT_FOUND
#define	LASSO_XML_ERROR_NODE_CONTENT_NOT_FOUND
#define	LASSO_XML_ERROR_ATTR_NOT_FOUND
#define	LASSO_XML_ERROR_ATTR_VALUE_NOT_FOUND
#define	LASSO_XML_ERROR_INVALID_FILE
#define	LASSO_XML_ERROR_OBJECT_CONSTRUCTION_FAILED
#define	LASSO_XML_ERROR_MISSING_NAMESPACE
#define	LASSO_DS_ERROR_SIGNATURE_NOT_FOUND

#define	LASSO_DS_ERROR_INVALID_SIGNATURE
#define	LASSO_DS_ERROR_SIGNATURE_TMPL_CREATION_FAILED
#define	LASSO_DS_ERROR_CONTEXT_CREATION_FAILED
#define	LASSO_DS_ERROR_PUBLIC_KEY_LOAD_FAILED
#define	LASSO_DS_ERROR_PRIVATE_KEY_LOAD_FAILED
#define	LASSO_DS_ERROR_CERTIFICATE_LOAD_FAILED
#define	LASSO_DS_ERROR_SIGNATURE_FAILED
#define	LASSO_DS_ERROR_KEYS_MNGR_CREATION_FAILED
#define	LASSO_DS_ERROR_KEYS_MNGR_INIT_FAILED
#define	LASSO_DS_ERROR_SIGNATURE_VERIFICATION_FAILED
#define	LASSO_DS_ERROR_CA_CERT_CHAIN_LOAD_FAILED
#define	LASSO_DS_ERROR_INVALID_SIGALG
#define	LASSO_DS_ERROR_DIGEST_COMPUTE_FAILED
#define	LASSO_DS_ERROR_SIGNATURE_TEMPLATE_NOT_FOUND
#define	LASSO_DS_ERROR_TOO_MUCH_REFERENCES
#define	LASSO_DS_ERROR_INVALID_REFERENCE_FOR_SAML
#define	LASSO_DS_ERROR_DECRYPTION_FAILED
#define	LASSO_DS_ERROR_ENCRYPTION_FAILED
#define	LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND
#define	LASSO_SERVER_ERROR_ADD_PROVIDER_FAILED
#define	LASSO_SERVER_ERROR_ADD_PROVIDER_PROTOCOL_MISMATCH
#define	LASSO_SERVER_ERROR_SET_ENCRYPTION_PRIVATE_KEY_FAILED
#define	LASSO_LOGOUT_ERROR_UNSUPPORTED_PROFILE
#define	LASSO_LOGOUT_ERROR_REQUEST_DENIED
#define	LASSO_LOGOUT_ERROR_FEDERATION_NOT_FOUND
#define	LASSO_LOGOUT_ERROR_UNKNOWN_PRINCIPAL
#define	LASSO_PROFILE_ERROR_INVALID_QUERY
#define	LASSO_PROFILE_ERROR_INVALID_POST_MSG
#define	LASSO_PROFILE_ERROR_INVALID_SOAP_MSG
#define	LASSO_PROFILE_ERROR_MISSING_REQUEST
#define	LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD
#define	LASSO_PROFILE_ERROR_INVALID_PROTOCOLPROFILE
#define	LASSO_PROFILE_ERROR_INVALID_MSG
#define	LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID
#define	LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE
#define	LASSO_PROFILE_ERROR_UNKNOWN_PROFILE_URL
#define	LASSO_PROFILE_ERROR_IDENTITY_NOT_FOUND
#define	LASSO_PROFILE_ERROR_FEDERATION_NOT_FOUND
#define	LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_FOUND
#define	LASSO_PROFILE_ERROR_BUILDING_QUERY_FAILED
#define	LASSO_PROFILE_ERROR_BUILDING_REQUEST_FAILED
#define	LASSO_PROFILE_ERROR_BUILDING_MESSAGE_FAILED
#define	LASSO_PROFILE_ERROR_BUILDING_RESPONSE_FAILED
#define	LASSO_PROFILE_ERROR_SESSION_NOT_FOUND
#define	LASSO_PROFILE_ERROR_BAD_IDENTITY_DUMP
#define	LASSO_PROFILE_ERROR_BAD_SESSION_DUMP
#define	LASSO_PROFILE_ERROR_MISSING_RESPONSE
#define	LASSO_PROFILE_ERROR_MISSING_STATUS_CODE
#define	LASSO_PROFILE_ERROR_MISSING_ARTIFACT
#define	LASSO_PROFILE_ERROR_MISSING_RESOURCE_OFFERING
#define	LASSO_PROFILE_ERROR_MISSING_SERVICE_DESCRIPTION
#define	LASSO_PROFILE_ERROR_MISSING_SERVICE_TYPE
#define	LASSO_PROFILE_ERROR_MISSING_ASSERTION
#define	LASSO_PROFILE_ERROR_MISSING_SUBJECT
#define	LASSO_PROFILE_ERROR_MISSING_NAME_IDENTIFIER
#define	LASSO_PROFILE_ERROR_INVALID_ARTIFACT
#define	LASSO_PROFILE_ERROR_MISSING_ENCRYPTION_PRIVATE_KEY

#define	LASSO_PROFILE_ERROR_STATUS_NOT_SUCCESS
#define	LASSO_PROFILE_ERROR_MISSING_ISSUER
#define	LASSO_PROFILE_ERROR_MISSING_SERVICE_INSTANCE
#define	LASSO_PROFILE_ERROR_MISSING_ENDPOINT_REFERENCE
#define	LASSO_PROFILE_ERROR_MISSING_ENDPOINT_REFERENCE_ADDRESS
#define	LASSO_PROFILE_ERROR_INVALID_ISSUER
#define	LASSO_PROFILE_ERROR_MISSING_SERVER
#define	LASSO_PROFILE_ERROR_UNKNOWN_PROVIDER
#define	LASSO_PROFILE_ERROR_CANNOT_VERIFY_SIGNATURE
#define	LASSO_PROFILE_ERROR_CANNOT_FIND_A_PROVIDER
#define	LASSO_PROFILE_ERROR_RESPONSE_DOES_NOT_MATCH_REQUEST
#define	LASSO_PROFILE_ERROR_INVALID_REQUEST
#define	LASSO_PROFILE_ERROR_INVALID_RESPONSE
#define	LASSO_PROFILE_ERROR_UNSUPPORTED_BINDING
#define	LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ
#define	LASSO_PARAM_ERROR_INVALID_VALUE
#define	LASSO_PARAM_ERROR_CHECK_FAILED
#define	LASSO_PARAM_ERROR_NON_INITIALIZED_OBJECT
#define	LASSO_LOGIN_ERROR_FEDERATION_NOT_FOUND
#define	LASSO_LOGIN_ERROR_CONSENT_NOT_OBTAINED
#define	LASSO_LOGIN_ERROR_INVALID_NAMEIDPOLICY
#define	LASSO_LOGIN_ERROR_REQUEST_DENIED
#define	LASSO_LOGIN_ERROR_INVALID_SIGNATURE
#define	LASSO_LOGIN_ERROR_UNSIGNED_AUTHN_REQUEST
#define	LASSO_LOGIN_ERROR_STATUS_NOT_SUCCESS
#define	LASSO_LOGIN_ERROR_UNKNOWN_PRINCIPAL
#define	LASSO_LOGIN_ERROR_NO_DEFAULT_ENDPOINT
#define	LASSO_LOGIN_ERROR_ASSERTION_REPLAY
#define	LASSO_LOGIN_ERROR_ASSERTION_DOES_NOT_MATCH_REQUEST_ID
#define	LASSO_DEFEDERATION_ERROR_MISSING_NAME_IDENTIFIER
#define	LASSO_SOAP_ERROR_REDIRECT_REQUEST_FAULT
#define	LASSO_SOAP_ERROR_MISSING_ENVELOPE
#define	LASSO_SOAP_ERROR_MISSING_HEADER
#define	LASSO_SOAP_ERROR_MISSING_BODY
#define	LASSO_SOAP_ERROR_MISSING_SOAP_FAULT_DETAIL
#define	LASSO_NAME_IDENTIFIER_MAPPING_ERROR_MISSING_TARGET_NAMESPACE
#define	LASSO_NAME_IDENTIFIER_MAPPING_ERROR_FORBIDDEN_CALL_ON_THIS_SI
#define	LASSO_NAME_IDENTIFIER_MAPPING_ERROR_MISSING_TARGET_IDENTIFIER
#define	LASSO_DATA_SERVICE_ERROR_UNREGISTERED_DST
#define	LASSO_WSF_PROFILE_ERROR_MISSING_CORRELATION
#define	LASSO_WSF_PROFILE_ERROR_MISSING_SECURITY
#define	LASSO_WSF_PROFILE_ERROR_MISSING_ASSERTION_ID
#define	LASSO_WSF_PROFILE_ERROR_MISSING_ENDPOINT
#define	LASSO_WSF_PROFILE_ERROR_SOAP_FAULT
#define	LASSO_WSF_PROFILE_ERROR_UNSUPPORTED_SECURITY_MECHANISM
#define	LASSO_WSF_PROFILE_ERROR_MISSING_DESCRIPTION
#define	LASSO_WSF_PROFILE_ERROR_MISSING_RESOURCE_ID
#define	LASSO_WSF_PROFILE_ERROR_MISSING_CREDENTIAL_REF
#define	LASSO_WSF_PROFILE_ERROR_INVALID_OR_MISSING_REFERENCE_TO_MESSA
#define	LASSO_DISCOVERY_ERROR_SVC_METADATA_REGISTER_FAILED
#define	LASSO_DISCOVERY_ERROR_SVC_METADATA_ASSOCIATION_ADD_FAILED
#define	LASSO_DISCOVERY_ERROR_MISSING_REQUESTED_SERVICE
#define	LASSO_DISCOVERY_ERROR_FAILED_TO_BUILD_ENDPOINT_REFERENCE
#define	LASSO_DST_ERROR_MISSING_SERVICE_DATA
#define	LASSO_DST_ERROR_QUERY_FAILED
#define	LASSO_DST_ERROR_QUERY_PARTIALLY_FAILED

#define	LASSO_DST_ERROR_MODIFY_FAILED
#define	LASSO_DST_ERROR_MODIFY_PARTIALLY_FAILED
#define	LASSO_DST_ERROR_NEW_DATA_MISSING
#define	LASSO_DST_ERROR_QUERY_NOT_FOUND
#define	LASSO_DST_ERROR_NO_DATA
#define	LASSO_DST_ERROR_MALFORMED_QUERY
#define	LASSO_REGISTRY_ERROR_KEY_EXISTS
#define	LASSO_PROVIDER_ERROR_MISSING_PUBLIC_KEY

Includes

```
#include <lasso/errors.h>
```

Description

Most functions in lasso return signed integer error codes. The convention is to give:

- a negative error code for programming or runtime recoverable errors,
- a positive error code for Liberty Alliance recoverable errors.

Beware that this convention is not always well followed.

Functions

Types and Values

LASSO_ERROR_UNDEFINED

```
#define LASSO_ERROR_UNDEFINED -1
```

Undefined error.

LASSO_ERROR_UNIMPLEMENTED

```
#define LASSO_ERROR_UNIMPLEMENTED -2
```

Unimplemented part of Lasso.

LASSO_ERROR_OUT_OF_MEMORY

```
#define LASSO_ERROR_OUT_OF_MEMORY -3
```

Out of memory

LASSO_ERROR_CAST_FAILED

```
#define LASSO_ERROR_CAST_FAILED -4
```

Expected GObject class was not found, cast failed

LASSO_XML_ERROR_NODE_NOT_FOUND

```
#define LASSO_XML_ERROR_NODE_NOT_FOUND -10
```

Unable to get child of element.

LASSO_XML_ERROR_NODE_CONTENT_NOT_FOUND

```
#define LASSO_XML_ERROR_NODE_CONTENT_NOT_FOUND -11
```

Unable to get content of element.

LASSO_XML_ERROR_ATTR_NOT_FOUND

```
#define LASSO_XML_ERROR_ATTR_NOT_FOUND -12
```

Unable to get attribute of element.

LASSO_XML_ERROR_ATTR_VALUE_NOT_FOUND

```
#define LASSO_XML_ERROR_ATTR_VALUE_NOT_FOUND -13
```

Unable to get attribute value of element.

LASSO_XML_ERROR_INVALID_FILE

```
#define LASSO_XML_ERROR_INVALID_FILE -14
```

Invalid XML file

LASSO_XML_ERROR_OBJECT_CONSTRUCTION_FAILED

```
#define LASSO_XML_ERROR_OBJECT_CONSTRUCTION_FAILED      -15
```

Construction of an object from an XML document failed.

LASSO_XML_ERROR_MISSING_NAMESPACE

```
#define LASSO_XML_ERROR_MISSING_NAMESPACE              -16
```

A namespace is missing.

LASSO_DS_ERROR_SIGNATURE_NOT_FOUND

```
#define LASSO_DS_ERROR_SIGNATURE_NOT_FOUND 101
```

Signature element not found.

LASSO_DS_ERROR_INVALID_SIGNATURE

```
#define LASSO_DS_ERROR_INVALID_SIGNATURE 102
```

Invalid signature.

LASSO_DS_ERROR_SIGNATURE_TMPL_CREATION_FAILED

```
#define LASSO_DS_ERROR_SIGNATURE_TMPL_CREATION_FAILED -103
```

LASSO_DS_ERROR_CONTEXT_CREATION_FAILED

```
#define LASSO_DS_ERROR_CONTEXT_CREATION_FAILED -104
```

Failed to create signature context.

LASSO_DS_ERROR_PUBLIC_KEY_LOAD_FAILED

```
#define LASSO_DS_ERROR_PUBLIC_KEY_LOAD_FAILED -105
```

Failed to load public key.

LASSO_DS_ERROR_PRIVATE_KEY_LOAD_FAILED

```
#define LASSO_DS_ERROR_PRIVATE_KEY_LOAD_FAILED -106
```

Failed to load private key.

LASSO_DS_ERROR_CERTIFICATE_LOAD_FAILED

```
#define LASSO_DS_ERROR_CERTIFICATE_LOAD_FAILED -107
```

Failed to load certificate.

LASSO_DS_ERROR_SIGNATURE_FAILED

```
#define LASSO_DS_ERROR_SIGNATURE_FAILED -108
```

Failed to sign the node.

LASSO_DS_ERROR_KEYS_MNGR_CREATION_FAILED

```
#define LASSO_DS_ERROR_KEYS_MNGR_CREATION_FAILED -109
```

Failed to create keys manager.

LASSO_DS_ERROR_KEYS_MNGR_INIT_FAILED

```
#define LASSO_DS_ERROR_KEYS_MNGR_INIT_FAILED -110
```

Failed to initialize keys manager.

LASSO_DS_ERROR_SIGNATURE_VERIFICATION_FAILED

```
#define LASSO_DS_ERROR_SIGNATURE_VERIFICATION_FAILED -111
```

Failed to verify signature.

LASSO_DS_ERROR_CA_CERT_CHAIN_LOAD_FAILED

```
#define LASSO_DS_ERROR_CA_CERT_CHAIN_LOAD_FAILED      -112
```

LASSO_DS_ERROR_INVALID_SIGALG

```
#define LASSO_DS_ERROR_INVALID_SIGALG -113
```

Invalid signature algorithm.

LASSO_DS_ERROR_DIGEST_COMPUTE_FAILED

```
#define LASSO_DS_ERROR_DIGEST_COMPUTE_FAILED          -114
```

Computation of an SHA1 digest failed.

LASSO_DS_ERROR_SIGNATURE_TEMPLATE_NOT_FOUND

```
#define LASSO_DS_ERROR_SIGNATURE_TEMPLATE_NOT_FOUND -115
```

Signature template has not been found.

LASSO_DS_ERROR_TOO_MUCH_REFERENCES

```
#define LASSO_DS_ERROR_TOO_MUCH_REFERENCES -116
```

SAML signature must contain only one reference

LASSO_DS_ERROR_INVALID_REFERENCE_FOR_SAML

```
#define LASSO_DS_ERROR_INVALID_REFERENCE_FOR_SAML -117
```

SAML signature reference must be to a Request, a Reponse or an Assertion ID attribute

LASSO_DS_ERROR_DECRYPTION_FAILED

```
#define LASSO_DS_ERROR_DECRYPTION_FAILED 118
```

Decryption of an encrypted node failed

LASSO_DS_ERROR_ENCRYPTION_FAILED

```
#define LASSO_DS_ERROR_ENCRYPTION_FAILED -119
```

Creation of an encrypted node failed

LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND

```
#define LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND -201
```

The identifier of a provider is unknown to LassoServer. To register a provider in a LassoServer object, you must use the methods lasso_server_add_provider() or lasso_server_add_provider_from_buffer().

LASSO_SERVER_ERROR_ADD_PROVIDER_FAILED

```
#define LASSO_SERVER_ERROR_ADD_PROVIDER_FAILED -202
```

Failed to add new provider.

LASSO_SERVER_ERROR_ADD_PROVIDER_PROTOCOL_MISMATCH

```
#define LASSO_SERVER_ERROR_ADD_PROVIDER_PROTOCOL_MISMATCH -203
```

Failed to add new provider (protocol mismatch). It means that you tried to add a provider supporting a protocol imcompatible with the protocol declared for your LassoServer, for example metadata for ID-FF 1.2 with metadata for SAML 2.0.

LASSO_SERVER_ERROR_SET_ENCRYPTION_PRIVATE_KEY_FAILED

```
#define LASSO_SERVER_ERROR_SET_ENCRYPTION_PRIVATE_KEY_FAILED 204
```

Failed to load encryption private key.

LASSO_LOGOUT_ERROR_UNSUPPORTED_PROFILE

```
#define LASSO_LOGOUT_ERROR_UNSUPPORTED_PROFILE -301
```

Unsupported protocol profile

LASSO_LOGOUT_ERROR_REQUEST_DENIED

```
#define LASSO_LOGOUT_ERROR_REQUEST_DENIED 302
```

Request denied by identity provider

LASSO_LOGOUT_ERROR_FEDERATION_NOT_FOUND

```
#define LASSO_LOGOUT_ERROR_FEDERATION_NOT_FOUND 303
```

Federation not found on logout

LASSO_LOGOUT_ERROR_UNKNOWN_PRINCIPAL

```
#define LASSO_LOGOUT_ERROR_UNKNOWN_PRINCIPAL 304
```

Unknown principal on logout

LASSO_PROFILE_ERROR_INVALID_QUERY

```
#define LASSO_PROFILE_ERROR_INVALID_QUERY -401
```

Invalid URL query

LASSO_PROFILE_ERROR_INVALID_POST_MSG

```
#define LASSO_PROFILE_ERROR_INVALID_POST_MSG -402
```

Invalid POST message

LASSO_PROFILE_ERROR_INVALID_SOAP_MSG

```
#define LASSO_PROFILE_ERROR_INVALID_SOAP_MSG -403
```

Invalid SOAP message

LASSO_PROFILE_ERROR_MISSING_REQUEST

```
#define LASSO_PROFILE_ERROR_MISSING_REQUEST -404
```

Missing request

LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD

```
#define LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD -405
```

Invalid HTTP method

LASSO_PROFILE_ERROR_INVALID_PROTOCOLPROFILE

```
#define LASSO_PROFILE_ERROR_INVALID_PROTOCOLPROFILE -406
```

Invalid protocol profile

LASSO_PROFILE_ERROR_INVALID_MSG

```
#define LASSO_PROFILE_ERROR_INVALID_MSG -407
```

Invalid message

LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID

```
#define LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID -408
```

ProviderID not found

LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE

```
#define LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE -409
```

Unsupported protocol profile

LASSO_PROFILE_ERROR_UNKNOWN_PROFILE_URL

```
#define LASSO_PROFILE_ERROR_UNKNOWN_PROFILE_URL -410
```

Unable to find Profile URL in metadata

LASSO_PROFILE_ERROR_IDENTITY_NOT_FOUND

```
#define LASSO_PROFILE_ERROR_IDENTITY_NOT_FOUND -411
```

Identity not found

LASSO_PROFILE_ERROR_FEDERATION_NOT_FOUND

```
#define LASSO_PROFILE_ERROR_FEDERATION_NOT_FOUND -412
```

Federation not found

LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_FOUND

```
#define LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_FOUND -413
```

Name identifier not found

LASSO_PROFILE_ERROR_BUILDING_QUERY_FAILED

```
#define LASSO_PROFILE_ERROR_BUILDING_QUERY_FAILED -414
```

Error building request QUERY url

LASSO_PROFILE_ERROR_BUILDING_REQUEST_FAILED

```
#define LASSO_PROFILE_ERROR_BUILDING_REQUEST_FAILED -415
```

Error building request object

LASSO_PROFILE_ERROR_BUILDING_MESSAGE_FAILED

```
#define LASSO_PROFILE_ERROR_BUILDING_MESSAGE_FAILED -416
```

Error building request message

LASSO_PROFILE_ERROR_BUILDING_RESPONSE_FAILED

```
#define LASSO_PROFILE_ERROR_BUILDING_RESPONSE_FAILED -417
```

Error building response object

LASSO_PROFILE_ERROR_SESSION_NOT_FOUND

```
#define LASSO_PROFILE_ERROR_SESSION_NOT_FOUND -418
```

Session not found

LASSO_PROFILE_ERROR_BAD_IDENTITY_DUMP

```
#define LASSO_PROFILE_ERROR_BAD_IDENTITY_DUMP -419
```

Failed to create identity from dump

LASSO_PROFILE_ERROR_BAD_SESSION_DUMP

```
#define LASSO_PROFILE_ERROR_BAD_SESSION_DUMP -420
```

Failed to create session from dump

LASSO_PROFILE_ERROR_MISSING_RESPONSE

```
#define LASSO_PROFILE_ERROR_MISSING_RESPONSE -421
```

Missing response

LASSO_PROFILE_ERROR_MISSING_STATUS_CODE

```
#define LASSO_PROFILE_ERROR_MISSING_STATUS_CODE -422
```

Missing status code

LASSO_PROFILE_ERROR_MISSING_ARTIFACT

```
#define LASSO_PROFILE_ERROR_MISSING_ARTIFACT -423
```

Missing SAML artifact

LASSO_PROFILE_ERROR_MISSING_RESOURCE_OFFERING

```
#define LASSO_PROFILE_ERROR_MISSING_RESOURCE_OFFERING 424
```

Missing ressource offering

LASSO_PROFILE_ERROR_MISSING_SERVICE_DESCRIPTION

```
#define LASSO_PROFILE_ERROR_MISSING_SERVICE_DESCRIPTION 425
```

Missing service description

LASSO_PROFILE_ERROR_MISSING_SERVICE_TYPE

```
#define LASSO_PROFILE_ERROR_MISSING_SERVICE_TYPE 426
```

Missing service type

LASSO_PROFILE_ERROR_MISSING_ASSERTION

```
#define LASSO_PROFILE_ERROR_MISSING_ASSERTION -427
```

When looking for an assertion we did not found it.

LASSO_PROFILE_ERROR_MISSING_SUBJECT

```
#define LASSO_PROFILE_ERROR_MISSING_SUBJECT -428
```

Missing subject

LASSO_PROFILE_ERROR_MISSING_NAME_IDENTIFIER

```
#define LASSO_PROFILE_ERROR_MISSING_NAME_IDENTIFIER -429
```

Missing name identifier

LASSO_PROFILE_ERROR_INVALID_ARTIFACT

```
#define LASSO_PROFILE_ERROR_INVALID_ARTIFACT -430
```

Invalid artifact

LASSO_PROFILE_ERROR_MISSING_ENCRYPTION_PRIVATE_KEY

```
#define LASSO_PROFILE_ERROR_MISSING_ENCRYPTION_PRIVATE_KEY -431
```

Found an encrypted element but encryption private key is not set

LASSO_PROFILE_ERROR_STATUS_NOT_SUCCESS

```
#define LASSO_PROFILE_ERROR_STATUS_NOT_SUCCESS -432
```

Status code is not success

LASSO_PROFILE_ERROR_MISSING_ISSUER

```
#define LASSO_PROFILE_ERROR_MISSING_ISSUER -433
```

Missing issuer

LASSO_PROFILE_ERROR_MISSING_SERVICE_INSTANCE

```
#define LASSO_PROFILE_ERROR_MISSING_SERVICE_INSTANCE -434
```

Missing service instance

LASSO_PROFILE_ERROR_MISSING_ENDPOINT_REFERENCE

```
#define LASSO_PROFILE_ERROR_MISSING_ENDPOINT_REFERENCE -435
```

Missing endpoint reference

LASSO_PROFILE_ERROR_MISSING_ENDPOINT_REFERENCE_ADDRESS

```
#define LASSO_PROFILE_ERROR_MISSING_ENDPOINT_REFERENCE_ADDRESS -436
```

Missing endpoint reference address

LASSO_PROFILE_ERROR_INVALID_ISSUER

```
#define LASSO_PROFILE_ERROR_INVALID_ISSUER -437
```

Assertion issuer is not the same as the requested issuer

LASSO_PROFILE_ERROR_MISSING_SERVER

```
#define LASSO_PROFILE_ERROR_MISSING_SERVER -438
```

No server object set in the profile

LASSO_PROFILE_ERROR_UNKNOWN_PROVIDER

```
#define LASSO_PROFILE_ERROR_UNKNOWN_PROVIDER 439
```

The issuer of the message is unknown to us

Parameters

Deprecated	Since 2.2.3

LASSO_PROFILE_ERROR_CANNOT_VERIFY_SIGNATURE

```
#define LASSO_PROFILE_ERROR_CANNOT_VERIFY_SIGNATURE 440
```

The profile cannot verify a signature on the message

LASSO_PROFILE_ERROR_CANNOT_FIND_A_PROVIDER

```
#define LASSO_PROFILE_ERROR_CANNOT_FIND_A_PROVIDER -441
```

Profile was called without a specific provider and we cannot find one.

LASSO_PROFILE_ERROR_RESPONSE_DOES_NOT_MATCH_REQUEST

```
#define LASSO_PROFILE_ERROR_RESPONSE_DOES_NOT_MATCH_REQUEST -442
```

Received response does not refer to the request sent

LASSO_PROFILE_ERROR_INVALID_REQUEST

```
#define LASSO_PROFILE_ERROR_INVALID_REQUEST 443
```

Received request is not of the expected type.

LASSO_PROFILE_ERROR_INVALID_RESPONSE

```
#define LASSO_PROFILE_ERROR_INVALID_RESPONSE 444
```

LASSO_PROFILE_ERROR_UNSUPPORTED_BINDING

```
#define LASSO_PROFILE_ERROR_UNSUPPORTED_BINDING 445
```

LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ

```
#define LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ -501
```

An object type provided as parameter is invalid or object is NULL.

LASSO_PARAM_ERROR_INVALID_VALUE

```
#define LASSO_PARAM_ERROR_INVALID_VALUE -502
```

A parameter value is invalid.

LASSO_PARAM_ERROR_CHECK_FAILED

```
#define LASSO_PARAM_ERROR_CHECK_FAILED -503
```

The error return location should be either NULL or contains a NULL error.

LASSO_PARAM_ERROR_NON_INITIALIZED_OBJECT

```
#define LASSO_PARAM_ERROR_NON_INITIALIZED_OBJECT        -504
```

The call failed because an argument is a partially-initialized object.

LASSO_LOGIN_ERROR_FEDERATION_NOT_FOUND

```
#define LASSO_LOGIN_ERROR_FEDERATION_NOT_FOUND 601
```

Federation not found on login

LASSO_LOGIN_ERROR_CONSENT_NOT_OBTAINED

```
#define LASSO_LOGIN_ERROR_CONSENT_NOT_OBTAINED         602
```

Consent of the principal was not obtained.

LASSO_LOGIN_ERROR_INVALID_NAMEIDPOLICY

```
#define LASSO_LOGIN_ERROR_INVALID_NAMEIDPOLICY -603
```

Invalid NameIDPolicy in lib:AuthnRequest

LASSO_LOGIN_ERROR_REQUEST_DENIED

```
#define LASSO_LOGIN_ERROR_REQUEST_DENIED 604
```

Request denied.

LASSO_LOGIN_ERROR_INVALID_SIGNATURE

```
#define LASSO_LOGIN_ERROR_INVALID_SIGNATURE            605
```

The signature of a message or of an assertion is invalid. That is badly computed or with an unknown key.

LASSO_LOGIN_ERROR_UNSIGNED_AUTHN_REQUEST

```
#define LASSO_LOGIN_ERROR_UNSIGNED_AUTHN_REQUEST       606
```

An unsigned authn request was received but the metadata specify that they must be signed.

LASSO_LOGIN_ERROR_STATUS_NOT_SUCCESS

```
#define LASSO_LOGIN_ERROR_STATUS_NOT_SUCCESS 607
```

Status code is not success

LASSO_LOGIN_ERROR_UNKNOWN_PRINCIPAL

```
#define LASSO_LOGIN_ERROR_UNKNOWN_PRINCIPAL 608
```

Unknown principal

LASSO_LOGIN_ERROR_NO_DEFAULT_ENDPOINT

```
#define LASSO_LOGIN_ERROR_NO_DEFAULT_ENDPOINT 609
```

No default endpoint

LASSO_LOGIN_ERROR_ASSERTION_REPLAY

```
#define LASSO_LOGIN_ERROR_ASSERTION_REPLAY 610
```

Assertion replay

LASSO_LOGIN_ERROR_ASSERTION_DOES_NOT_MATCH_REQUEST_ID

```
#define LASSO_LOGIN_ERROR_ASSERTION_DOES_NOT_MATCH_REQUEST_ID 611
```

If inResponseTo attribute is present, a matching request must be present too in the LassoLogin object

LASSO_DEFEDERATION_ERROR_MISSING_NAME_IDENTIFIER

```
#define LASSO_DEFEDERATION_ERROR_MISSING_NAME_IDENTIFIER -700
```

Name identifier not found in request

LASSO_SOAP_ERROR_REDIRECT_REQUEST_FAULT

```
#define LASSO_SOAP_ERROR_REDIRECT_REQUEST_FAULT 800
```

A SOAP Fault containing a Redirect Request was received

LASSO_SOAP_ERROR_MISSING_ENVELOPE

```
#define LASSO_SOAP_ERROR_MISSING_ENVELOPE -801
```

Missing SOAP envelope

LASSO_SOAP_ERROR_MISSING_HEADER

```
#define LASSO_SOAP_ERROR_MISSING_HEADER -802
```

Missing SOAP header

LASSO_SOAP_ERROR_MISSING_BODY

```
#define LASSO_SOAP_ERROR_MISSING_BODY -803
```

Missing SOAP body

LASSO_SOAP_ERROR_MISSING_SOAP_FAULT_DETAIL

```
#define LASSO_SOAP_ERROR_MISSING_SOAP_FAULT_DETAIL -804
```

Missing SOAP fault detail

LASSO_NAME_IDENTIFIER_MAPPING_ERROR_MISSING_TARGET_NAMESPACE

```
#define LASSO_NAME_IDENTIFIER_MAPPING_ERROR_MISSING_TARGET_NAMESPACE -900
```

Target name space not found

LASSO_NAME_IDENTIFIER_MAPPING_ERROR_FORBIDDEN_CALL_ON_THIS_SIDE

```
#define LASSO_NAME_IDENTIFIER_MAPPING_ERROR_FORBIDDEN_CALL_ON_THIS_SIDE -901
```

LASSO_NAME_IDENTIFIER_MAPPING_ERROR_MISSING_TARGET_IDENTIFIER

```
#define LASSO_NAME_IDENTIFIER_MAPPING_ERROR_MISSING_TARGET_IDENTIFIER -902
```

LASSO_DATA_SERVICE_ERROR_UNREGISTERED_DST

```
#define LASSO_DATA_SERVICE_ERROR_UNREGISTERED_DST    -1000
```

LASSO_WSF_PROFILE_ERROR_MISSING_CORRELATION

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_CORRELATION -1100
```

Correlation SOAP Header is missing

LASSO_WSF_PROFILE_ERROR_MISSING_SECURITY

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_SECURITY -1101
```

Security SOAP Header is missing

LASSO_WSF_PROFILE_ERROR_MISSING_ASSERTION_ID

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_ASSERTION_ID -1102
```

AssertionID attribute is missing

LASSO_WSF_PROFILE_ERROR_MISSING_ENDPOINT

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_ENDPOINT -1103
```

Cannot find an WSP endpoint for the ID-WSF service

LASSO_WSF_PROFILE_ERROR_SOAP_FAULT

```
#define LASSO_WSF_PROFILE_ERROR_SOAP_FAULT 1104
```

SOAP ID-WSF binding returned a SOAP fault

LASSO_WSF_PROFILE_ERROR_UNSUPPORTED_SECURITY_MECHANISM

```
#define LASSO_WSF_PROFILE_ERROR_UNSUPPORTED_SECURITY_MECHANISM 1105
```

The specified security mechanism is not supported by lasso ID-WSF library

LASSO_WSF_PROFILE_ERROR_MISSING_DESCRIPTION

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_DESCRIPTION -1106
```

No ID-WSF web Service description could be found for the current security mechanism

LASSO_WSF_PROFILE_ERROR_MISSING_RESOURCE_ID

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_RESOURCE_ID -1107
```

The necessary ResourceID or EncryptedResourceID for calling an ID-WSF service is missing.

LASSO_WSF_PROFILE_ERROR_MISSING_CREDENTIAL_REF

```
#define LASSO_WSF_PROFILE_ERROR_MISSING_CREDENTIAL_REF -1108
```

WS-Security SAML Token secmech needs a CredentialRef

LASSO_WSF_PROFILE_ERROR_INVALID_OR_MISSING_REFERENCE_TO_MESSAGE_ID

```
#define LASSO_WSF_PROFILE_ERROR_INVALID_OR_MISSING_REFERENCE_TO_MESSAGE_ID -1109
```

refToMessageID attribute of the Corrrelation header does not match the SOAP request

LASSO_DISCOVERY_ERROR_SVC_METADATA_REGISTER_FAILED

```
#define LASSO_DISCOVERY_ERROR_SVC_METADATA_REGISTER_FAILED -1200
```

Service metadata registration failed

LASSO_DISCOVERY_ERROR_SVC_METADATA_ASSOCIATION_ADD_FAILED

```
#define LASSO_DISCOVERY_ERROR_SVC_METADATA_ASSOCIATION_ADD_FAILED -1201
```

Service metadata association failed

LASSO_DISCOVERY_ERROR_MISSING_REQUESTED_SERVICE

```
#define LASSO_DISCOVERY_ERROR_MISSING_REQUESTED_SERVICE -1202
```

Missing requested service

LASSO_DISCOVERY_ERROR_FAILED_TO_BUILD_ENDPOINT_REFERENCE

```
#define LASSO_DISCOVERY_ERROR_FAILED_TO_BUILD_ENDPOINT_REFERENCE -1203
```

Failed to build Endpoint Reference

LASSO_DST_ERROR_MISSING_SERVICE_DATA

```
#define LASSO_DST_ERROR_MISSING_SERVICE_DATA -1300
```

Missing service data

LASSO_DST_ERROR_QUERY_FAILED

```
#define LASSO_DST_ERROR_QUERY_FAILED -1301
```

Query failed

LASSO_DST_ERROR_QUERY_PARTIALLY_FAILED

```
#define LASSO_DST_ERROR_QUERY_PARTIALLY_FAILED -1302
```

Query partially failed : some items were correctly processed

LASSO_DST_ERROR_MODIFY_FAILED

```
#define LASSO_DST_ERROR_MODIFY_FAILED -1303
```

Modify failed

LASSO_DST_ERROR_MODIFY_PARTIALLY_FAILED

```
#define LASSO_DST_ERROR_MODIFY_PARTIALLY_FAILED -1304
```

Modify partially failed : some items were correctly processed

LASSO_DST_ERROR_NEW_DATA_MISSING

```
#define LASSO_DST_ERROR_NEW_DATA_MISSING -1305
```

Missing new data

LASSO_DST_ERROR_QUERY_NOT_FOUND

```
#define LASSO_DST_ERROR_QUERY_NOT_FOUND -1306
```

Looked query is not found

LASSO_DST_ERROR_NO_DATA

```
#define LASSO_DST_ERROR_NO_DATA -1307
```

No data or no data for the designated query item in the query response

LASSO_DST_ERROR_MALFORMED_QUERY

```
#define LASSO_DST_ERROR_MALFORMED_QUERY -1308
```

QueryObject is malformed

LASSO_REGISTRY_ERROR_KEY_EXISTS

```
#define LASSO_REGISTRY_ERROR_KEY_EXISTS -1400
```

Key alreadys exists in the registry

LASSO_PROVIDER_ERROR_MISSING_PUBLIC_KEY

```
#define LASSO_PROVIDER_ERROR_MISSING_PUBLIC_KEY -1500
```

The provider has no known public key

LassoNode is the base class for all Lasso classes, it gives XML serialization and deserialization support to all of them.

3.8 LassoNode

LassoNode — Base class for all Lasso objects

Functions

LassoNode *	lasso_node_new ()
LassoNode *	lasso_node_new_from_dump ()
LassoNode *	lasso_node_new_from_soap ()
LassoNode *	lasso_node_new_from_xmlNode ()
void	lasso_node_cleanup_original_xmlnodes ()
void	lasso_node_destroy ()
char *	lasso_node_dump ()
char *	lasso_node_debug ()
char *	lasso_node_export_to_base64 ()
char *	lasso_node_export_to_query ()
char *	lasso_node_export_to_query_with_password ()
char *	lasso_node_export_to_soap ()
char *	lasso_node_export_to_soap_with_headers ()
gchar *	lasso_node_export_to_xml ()
char *	lasso_node_export_to_paos_request ()
char *	lasso_node_export_to_paos_request_full ()
char *	lasso_node_export_to_ecp_soap_response ()
xmlNode *	lasso_node_get_xmlNode ()
const char *	lasso_node_get_name ()
xmlNode *	lasso_node_get_original_xmlnode ()
void	lasso_node_set_original_xmlnode ()
void	lasso_node_set_custom_namespace ()
void	lasso_node_set_custom_nodename ()
const char *	lasso_node_get_namespace ()
LassoMessageFormat	lasso_node_init_from_message ()
gboolean	lasso_node_init_from_query ()
lasso_error_t	lasso_node_init_from_xml ()
const char *	lasso_strerror ()

Types and Values

enum	LassoMessageFormat
enum	LassoSignatureType
enum	LassoSignatureMethod
	LassoNodeClassData
struct	LassoNode

Description

LassoNode is the base class for Lasso objects; just a step over GObject as defined in glib.

Functions

lasso_node_new ()

```
LassoNode~*
lasso_node_new (void);
```

Creates a new LassoNode.

Returns

a newly created LassoNode object

lasso_node_new_from_dump ()

```
LassoNode~*
lasso_node_new_from_dump (const char *dump);
```

Restores the *dump* to a new LassoNode subclass.

Parameters

dump	XML object dump

Returns

a newly created object; or NULL if an error occured.

lasso_node_new_from_soap ()

```
LassoNode~*
lasso_node_new_from_soap (const char *soap);
```

Parses SOAP message and creates a new Lasso object with the right class.

Parameters

soap	the SOAP message

Returns

node if success; NULL otherwise

lasso_node_new_from_xmlNode ()

```
LassoNode~*
lasso_node_new_from_xmlNode (xmlNode *node);
```

Builds a new LassoNode from an xmlNode.

Parameters

node	an xmlNode

Returns

a new node

lasso_node_cleanup_original_xmlnodes ()

```
void
lasso_node_cleanup_original_xmlnodes (LassoNode *node);
```

`node` : a LassoNode

Traverse the LassoNode tree starting at Node and remove keeped xmlNode if one is found.

Returns

None

lasso_node_destroy ()

```
void
lasso_node_destroy (LassoNode *node);
```

Destroys the LassoNode.

Parameters

node	a LassoNode

lasso_node_dump ()

```
char~*
lasso_node_dump (LassoNode *node);
```

Dumps `node` . All datas in object are dumped in an XML format.

Parameters

node	a LassoNode

Returns

a full XML dump of `node` . The string must be freed by the caller.

[transfer full]

lasso_node_debug ()

```
char~*
lasso_node_debug (LassoNode *node,
                  int level);
```

Create a debug dump for `node` , it is pretty printed so any contained signature will be uncheckable.

Parameters

node	a LassoNode	
level	the indentation depth, i.e. the depth of the last nodes to be indented.	*[default 10]*

Returns

a full indented and so human readable dump of *node* . The string must be freed by the caller.

[transfer full]

lasso_node_export_to_base64 ()

```
char~*
lasso_node_export_to_base64 (LassoNode *node);
```

Exports *node* to a base64-encoded message.

Parameters

node	a LassoNode	

Returns

a base64-encoded export of *node* . The string must be freed by the caller.

lasso_node_export_to_query ()

```
char~*
lasso_node_export_to_query (LassoNode *node,
                            LassoSignatureMethod sign_method,
                            const char *private_key_file);
```

Exports *node* to a HTTP query string. If *private_key_file* is NULL, query won't be signed.

Parameters

node	a LassoNode	
sign_method	the Signature transform method.	*[default 1]*
private_key_file	the path to the private key (may be NULL).	*[allow-none]*

Returns

a HTTP query export of *node* . The string must be freed by the caller.

lasso_node_export_to_query_with_password ()

```
char~*
lasso_node_export_to_query_with_password
                                (LassoNode *node,
                                                LassoSignatureMethod sign_method,
                                                const char *private_key_file,
                                                const char *private_key_file_password);
```

Exports *node* to a HTTP query string. If *private_key_file* is NULL, query won't be signed.

Parameters

node	a LassoNode	
sign_method	the Signature transform method.	*[default 1]*
private_key_file	the path to the private key (may be NULL).	*[allow-none]*
private_key_file_password	the password needed to decrypt the private key.	*[allow-none]*

Returns

a HTTP query export of *node* . The string must be freed by the caller.

lasso_node_export_to_soap ()

```
char~*
lasso_node_export_to_soap (LassoNode *node);
```

Exports *node* to a SOAP message.

Parameters

node	a LassoNode	

Returns

a SOAP export of *node* . The string must be freed by the caller.

lasso_node_export_to_soap_with_headers ()

```
char~*
lasso_node_export_to_soap_with_headers
                               (LassoNode *node,
                                GList *headers);
```

Exports *node* to a SOAP message. The *node* becomes the SOAP body. each header in the headers list is added to the SOAP header if non-NULL. *headers* is permitted to be an empty list (e.g. NULL).

Example 3.1 Create SOAP envelope with variable number of header nodes

You need to form a SOAP message with authn_request as the body and paos_request, ecp_request and ecp_relaystate as SOAP header elements. It is possible one or more of these may be NULL and should be skipped.

```
char *text = NULL;
LassoNode *paos_request = NULL;
LassoNode *ecp_request = NULL;
LassoNode *ecp_relaystate = NULL;
```

```
GList *headers = NULL;

paos_request = lasso_paos_request_new(responseConsumerURL, message_id);
ecp_request = lasso_ecp_request_new(issuer, is_passive, provider_name, idp_list);

lasso_list_add_new_gobject(headers, paos_request);
lasso_list_add_new_gobject(headers, ecp_request);
lasso_list_add_new_gobject(headers, ecp_relaystate);

text = lasso_node_export_to_soap_with_headers(node, headers);

lasso_release_list_of_gobjects(headers);
```

Parameters

node	a LassoNode, becomes the SOAP body	
headers	GList of LassNode.	*[allow-none]*

Returns

a SOAP export of `node` . The string must be freed by the caller.

lasso_node_export_to_xml ()

```
gchar~*
lasso_node_export_to_xml (LassoNode *node);
```

Exports `node` to an xml message.

Parameters

node	a LassoNode	

Returns

an xml export of `node` . The string must be freed by the caller.

lasso_node_export_to_paos_request ()

```
char~*
lasso_node_export_to_paos_request (LassoNode *node,
                                   const char *issuer,
                                   const char *responseConsumerURL,
                                   const char *relay_state);
```

Exports `node` to a PAOS message.

Deprecated, use lasso_node_export_to_paos_request_full() instead

Parameters

node	a LassoNode

Returns

a PAOS export of *node* . The string must be freed by the caller.

lasso_node_export_to_paos_request_full ()

```
char~*
lasso_node_export_to_paos_request_full
                               (LassoNode *node,
                                const char *issuer,
                                const char *responseConsumerURL,
                                const char *message_id,
                                const char *relay_state,
                                gboolean is_passive,
                                gchar *provider_name,
                                LassoSamlp2IDPList *idp_list);
```

Creates a new SOAP message. The SOAP headers include a PaosRequst, a EcpRequest and optionally a EcpRelayState. The SOAP body contains the *node* parameters.

Parameters

message_id	.	*[allow-none]*
relay_state	.	*[allow-none]*
provider_name	.	*[allow-none]*
idp_list	.	*[allow-none]*

Returns

string containing a PAOS request. The string must be freed by the caller.

lasso_node_export_to_ecp_soap_response ()

```
char~*
lasso_node_export_to_ecp_soap_response
                               (LassoNode *node,
                                const char *assertionConsumerURL);
```

Exports *node* to a ECP SOAP message.

Parameters

node	a LassoNode

Returns

a ECP SOAP export of *node* . The string must be freed by the caller.

lasso_node_get_xmlNode ()

```
xmlNode~*
lasso_node_get_xmlNode (LassoNode *node,
                        gboolean lasso_dump);
```

Builds an XML representation of `node` .

Parameters

node	a LassoNode	
lasso_dump	whether to include lasso-specific nodes	

Returns

a new xmlNode. It must be freed by the caller.

lasso_node_get_name ()

```
const char~*
lasso_node_get_name (LassoNode *node);
```

Return the XML element name for this object, the one that would be used in the XML dump of this object.

Parameters

node	a LassoNode	

Returns

the name of the object, the value must not be stored.

lasso_node_get_original_xmlnode ()

```
xmlNode~*
lasso_node_get_original_xmlnode (LassoNode *node);
```

Retrieve the original xmlNode eventually associated to this LassoNode.

Parameters

node	a LassoNode	

Returns

an xmlNodePtr or NULL.

[transfer none]

lasso_node_set_original_xmlnode ()

```
void
lasso_node_set_original_xmlnode (LassoNode *node,
                                 xmlNode *xmlnode);
```

Set the underlying XML representation of the object.

Parameters

node	the LassoNode object	
xmlnode	an xmlNode	

lasso_node_set_custom_namespace ()

```
void
lasso_node_set_custom_namespace (LassoNode *node,
                                 const char *prefix,
                                 const char *href);
```

Set a custom namespace for an object instance, use it with object existing a lot of revision of the nearly same namespace.

Parameters

node	a LassoNode object	
prefix	the prefix to use for the definition	
href	the URI of the namespace	

lasso_node_set_custom_nodename ()

```
void
lasso_node_set_custom_nodename (LassoNode *node,
                                const char *nodename);
```

Set a custom nodename for an object instance, use it with object implement a schema type and not a real element.

Parameters

node	a LassoNode object	
nodename	the name to use for the node	

lasso_node_get_namespace ()

```
const char~*
lasso_node_get_namespace (LassoNode *node);
```

lasso_node_init_from_message ()

```
LassoMessageFormat
lasso_node_init_from_message (LassoNode *node,
                              const char *message);
```

Parses *message* and initialiazes *node* fields with data from it. Message type may be base64, SOAP, XML or query string, correct type is found automatically.

Parameters

node	a LassoNode (or derived class)	
message	a Liberty message	

Returns

a LassoMessageFormat value.

lasso_node_init_from_query ()

```
gboolean
lasso_node_init_from_query (LassoNode *node,
                            const char *query);
```

Initialiazes *node* fields with data from *query* string.

Parameters

node	a LassoNode (or derived class)	
query	the query string	

Returns

TRUE if success

lasso_node_init_from_xml ()

```
lasso_error_t
lasso_node_init_from_xml (LassoNode *node,
                          xmlNode *xmlnode);
```

Initialiazes *node* fields with data from *xmlnode* XML node.

Parameters

node	a LassoNode (or derived class)	
xmlnode	the libxml2 node	

Returns

0 on success; or a negative value otherwise.

lasso_strerror ()

```
const char~*
lasso_strerror (int error_code);
```

Convert an error code from a lasso fuction to a human readable string.

Parameters

error_code	a gint error code returned by a lasso function

Returns

a static string.

Types and Values

enum LassoMessageFormat

Return code for lasso_node_init_from_message; it describes the type of the message that was passed to that function.

Members

LASSO_MESSAGE_FORMAT_XSCHEMA_ERROR	
LASSO_MESSAGE_FORMAT_ERROR	error while determining format
LASSO_MESSAGE_FORMAT_UNKNOWN	unknown format
LASSO_MESSAGE_FORMAT_XML	XML
LASSO_MESSAGE_FORMAT_BASE64	base-64 encoded
LASSO_MESSAGE_FORMAT_QUERY	query string
LASSO_MESSAGE_FORMAT_SOAP	SOAP

enum LassoSignatureType

Signature type.

Members

LASSO_SIGNATURE_TYPE_NONE	no signature	
LASSO_SIGNATURE_TYPE_SIMPLE	sign with the private key, copy the public part in the signature.	
LASSO_SIGNATURE_TYPE_WITHX509	sign with the private key, copy the associated certificat in the signature.	
LASSO_SIGNATURE_TYPE_LAST		

enum LassoSignatureMethod

Signature method.

Members

LASSO_SIGNATURE_METHOD_NONE		

LASSO_SIGNATURE_METHOD_RSA_SHA1	sign using a RSA private key
LASSO_SIGNATURE_METHOD_DSA_SHA1	sign using a DSA private key
LASSO_SIGNATURE_METHOD_HMAC_SHA1	sign using a HMAC-SHA1 secret key
LASSO_SIGNATURE_METHOD_RSA_SHA256	sign using a RSA private key and use SHA256 digest
LASSO_SIGNATURE_METHOD_HMAC_SHA256	sign using a HMAC-SHA256 secret key

LASSO_SIGNATURE_METHOD_RSA_SHA384		sign using a RSA private key and use SHA384 digest
LASSO_SIGNATURE_METHOD_HMAC_SHA384		sign using a HMAC-SHA384 secret key
LASSO_SIGNATURE_METHOD_RSA_SHA512		sign using a RSA private key and use SHA521 digest
LASSO_SIGNATURE_METHOD_HMAC_SHA512		sign using a HMAC-SHA512 secret key
LASSO_SIGNATURE_METHOD_LAST		

LassoNodeClassData

```
typedef struct _LassoNodeClassData LassoNodeClassData;
```

struct LassoNode

```
struct LassoNode {
  GObject parent;
};
```

Base type for all XML contents, or for object using serialization to XML.

The LassoMiscTextNode allows to represent miscellenaous nodes for whose no mapping to a specific GObjectClass exists.

3.9 Base Strings

Base Strings — General strings constants for Lasso

Types and Values

#define	LASSO_PYTHON_HREF
#define	LASSO_LASSO_PREFIX
#define	LASSO_LASSO_HREF

Includes

```
#include <lasso/xml/strings.h>
```

Description

Functions

Types and Values

LASSO_PYTHON_HREF

```
#define LASSO_PYTHON_HREF "http://www.entrouvert.org/namespaces/python/0.0"
```

LASSO_LASSO_PREFIX

```
#define LASSO_LASSO_PREFIX "lasso"
```

Preferred prefix for the lasso internal serialization format namespace.

LASSO_LASSO_HREF

```
#define LASSO_LASSO_HREF   "http://www.entrouvert.org/namespaces/lasso/0.0"
```

Namespace for Lasso internal serialization format

3.10 registry

registry — Class to store a mapping of qualified names (QName) to other qualified names.

Functions

const char *	(*LassoRegistryTranslationFunction) ()
lasso_error_t	lasso_registry_default_add_direct_mapping ()
lasso_error_t	lasso_registry_default_add_functional_mapping ()
const char *	lasso_registry_default_get_mapping ()

Description

A qualified name is a name or a string in the context of another name, or namespace. This object implement a function of a tuple (namespace, name, namespace) to a name. For the moment there is no need to enumerate all tuples (namespace, name) pair given a base pair, i.e. a function from tuple (namespace, name) to a list of tuples (namespace,name).

We support two kinds of mapping:

- you can give a direct mapping between two QName,
- or you can give a function that will manage mapping between one namespace and another one.

For internal use inside lasso we define the following namespaces:

- LASSO_LASSO_HREF and,
- LASSO_PYTHON_HREF.

For functional mappings the mapping function must return constant strings created using g_intern_string() or using g_type_name().

Functions

LassoRegistryTranslationFunction ()

```
const char~*
(*LassoRegistryTranslationFunction) (const char *from_namespace,
                                     const char *from_name,
                                     const char *to_namespace);
```

lasso_registry_default_add_direct_mapping ()

```
lasso_error_t
lasso_registry_default_add_direct_mapping
                               (const char *from_namespace,
                                const char *from_name,
                                const char *to_namespace,
                                const char *to_name);
```

Add a new mapping from a QName to a QName.

Parameters

from_namespace	the namespace of the mapped QName	
from_name	the name of the mapped QName	
to_namespace	the namepsace of the mapped to QName	
to_name	the name of the mapped to QName	

Returns

0 if successfull, LASSO_REGISTRY_ERROR_KEY_EXISTS if this mapping is already registered, LASSO_PARAM_ERROR_INVAL if one the argument is invalid.

lasso_registry_default_add_functional_mapping ()

```
lasso_error_t
lasso_registry_default_add_functional_mapping
                               (const char *from_namespace,
                                const char *to_namespace,
                                LassoRegistryTranslationFunction translation_function);
```

from_namespace : URI of the source namespace *to_namespace* : URI of the destination namespace *translation_function* : a function mapping string to string from the first namespace to the second one

Register a new mapping from from_namesapce to to_namespace using the translation_function into the default mapping. This functions is not forced to return a value for any string, it can return NULL.

Returns

0 if successfull, LASSO_REGISTRY_ERROR_KEY_EXISTS if this mapping is already registered, LASSO_PARAM_ERROR_INVAL if one the argument is invalid.

lasso_registry_default_get_mapping ()

```
const char~*
lasso_registry_default_get_mapping (const char *from_namespace,
                                    const char *from_name,
                                    const char *to_namespace);
```

Retrieve the name of the QName in the namespace to_namespace that maps the QName from_namespace:from_name.

Parameters

from_namespace	the namespace of the mapped QName	
from_name	the name of the mapped QName	
to_namespace	the namepsace of the mapped to QName	

Returns

the name string of the QName or NULL if no mapping exists.

3.11 LassoMiscTextNode

LassoMiscTextNode —

Functions

LassoNode *	lasso_misc_text_node_new ()
LassoMiscTextNode *	lasso_misc_text_node_new_with_string ()
LassoMiscTextNode *	lasso_misc_text_node_new_with_xml_node ()
xmlNode *	lasso_misc_text_node_get_xml_content ()
void	lasso_misc_text_node_set_xml_content ()

Types and Values

struct	LassoMiscTextNode

Description

Functions

lasso_misc_text_node_new ()

```
LassoNode~*
lasso_misc_text_node_new (void);
```

Creates a new LassoMiscTextNode object.

Returns

a newly created LassoMiscTextNode object

lasso_misc_text_node_new_with_string ()

```
LassoMiscTextNode~*
lasso_misc_text_node_new_with_string (const char *content);
```

Creates a new LassoMiscTextNode object and initializes it with `content` . Beware that no nodename, so it would create a text child, not an element.

Parameters

content	the content of newly created LassoMiscTextNode

Returns

a newly created LassoMiscTextNode object

lasso_misc_text_node_new_with_xml_node ()

```
LassoMiscTextNode~*
lasso_misc_text_node_new_with_xml_node
                              (xmlNode *xml_node);
```

Creates a new LassoMiscTextNode object and initialize it with *xml_node* .

Parameters

xml_node	an xmlNode

Returns

a newly created LassoMiscTextNode object

lasso_misc_text_node_get_xml_content ()

```
xmlNode~*
lasso_misc_text_node_get_xml_content (LassoMiscTextNode *misc_text_node);
```

Return the xml content in this node.

Parameters

misc_text_node	a LassoMiscTextNode

Returns

an xmlNode or NULL.

[transfer none]

lasso_misc_text_node_set_xml_content ()

```
void
lasso_misc_text_node_set_xml_content (LassoMiscTextNode *misc_text_node,
                                      xmlNode *node);
```

Set the xml content of this LassoMiscTextNode

Parameters

misc_text_node	a LassoMiscTextNode

Types and Values

struct LassoMiscTextNode

```
struct LassoMiscTextNode {
  LassoNode parent;

  /* elements */
  char *content;

  char *name;
  char *ns_href;
  char *ns_prefix;
  gboolean text_child;
};
```

Chapter 4

Identity Federation Framework - ID-FF 1.2 profiles

4.1 LassoLogin

LassoLogin — Single Sign-On and Federation Profile

Functions

LassoLogin *	lasso_login_new ()
LassoLogin *	lasso_login_new_from_dump ()
lasso_error_t	lasso_login_accept_sso ()
lasso_error_t	lasso_login_build_artifact_msg ()
lasso_error_t	lasso_login_build_assertion ()
lasso_error_t	lasso_login_build_authn_request_msg ()
lasso_error_t	lasso_login_build_authn_response_msg ()
lasso_error_t	lasso_login_build_request_msg ()
lasso_error_t	lasso_login_build_response_msg ()
void	lasso_login_destroy ()
gchar *	lasso_login_dump ()
LassoNode *	lasso_login_get_assertion ()
lasso_error_t	lasso_login_init_authn_request ()
lasso_error_t	lasso_login_init_idp_initiated_authn_request ()
lasso_error_t	lasso_login_init_request ()
gboolean	lasso_login_must_ask_for_consent ()
gboolean	lasso_login_must_authenticate ()
lasso_error_t	lasso_login_process_authn_request_msg ()
lasso_error_t	lasso_login_process_authn_response_msg ()
lasso_error_t	lasso_login_process_paos_response_msg ()
lasso_error_t	lasso_login_process_request_msg ()
lasso_error_t	lasso_login_process_response_msg ()
lasso_error_t	lasso_login_validate_request_msg ()

Types and Values

struct	LassoLogin

Description

The Single Sign On process allows a user to log in once to an identity provider (IdP), and to be then transparently loged in to the required service providers (SP) belonging to the IP "circle of trust". Subordinating different identities of the same user within a circle of trust to a unique IP is called "Identity Federation". The liberty Alliance specifications allows, thanks to this federation, strong and unique authentication coupled with control by the user of his personal informations. The explicit user agreement is necessary before proceeding to Identity Federation.

The service provider must implement the following process:

- creating an authentication request with lasso_login_init_authn_request();
- sending it to the identity provider with lasso_login_build_authn_request_msg();
- receiving and processing the answer:
  - either an authentication response with lasso_login_process_authn_response_msg()
  - or an artifact with lasso_login_init_request() then sending the request to the IdP with lasso_login_build_request_msg() and processing the new answer with lasso_login_process_response_msg().

Our first example shows how to initiate a request toward an ID-FF 1.2 or SAML 2.0 identity provider. It supposes that we already initialized a LassoServer object with the metadatas or our provider (and its private key if we want to sign the request), and that we added the metadatas of the targetted IdP with the method lasso_server_add_provider().

Example 4.1 Service Provider Login URL

```
LassoLogin *login;
int rc; // hold return codes

login = lasso_login_new(server);
rc = lasso_login_init_authn_request(login, "http://identity-provider-id/",
                LASSO_HTTP_METHOD_REDIRECT);
if (rc != 0) {
  ... // handle errors, most of them are related to bad initialization
}

// customize AuthnRequest
// protocolProfile is the protocolProfile of the provider http://identity-provider-id/
if (protocolProfile == LASSO_LIBERTY_1_2) {
        LassoLibAuthnRequest *request = LASSO_LIB_AUTHN_REQUEST(LASSO_PROFILE(login)-> ↩
            request);
        request->NameIDPolicy = strdup(LASSO_LIB_NAMEID_POLICY_TYPE_FEDERATED);
        request->ForceAuthn = TRUE;
        request->IsPassive = FALSE;
        // tell the IdP how to return the response
        request->ProtocolProfile = strdup(LASSO_LIB_PROTOCOL_PROFILE_BRWS_ART);
} else if (protocolProfile == LASSO_SAML_2_0) {
        LassoSamlp2AuthnRequest *request = LASSO_SAMLP2_AUTHN_REQUEST(LASSO_PROFILE(login) ↩
            ->request);
        if (request->NameIDPolicy->Format) {
                g_free(request->NameIDPolicy->Format);
        }
        request->NameIDPolicy->Format = g_strdup(LASSO_NAME_IDENTIFIER_FORMAT_PERSISTENT);
        // Allow creation of new federation
        //
        request->NameIDPolicy->AllowCreate = 1;
        request->ForceAuthn = TRUE;
        request->IsPassive = FALSE;
        // tell the IdP how to return the response
        if (request->ProtocolBinding) {
                 g_free(request->ProtocolBinding);
        }
        // here we expect an artifact response, it could be post, redirect or PAOS.
        request->ProtocolBinding = g_strdup(LASSO_SAML2_METADATA_BINDING_ARTIFACT);
   }
// Lasso will choose whether to sign the request by looking at the IdP
// metadatas and at our metadatas, but you can always force him to sign or to
// not sign using the method lasso_profile_set_signature_hint() on the
// LassoLogin object.

rc = lasso_login_build_authn_request_msg(login);
if (rc != 0) {
      .... // handle errors
      // could be that the requested binding (POST, Redirect, etc..) is not supported ( ↩
          LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE)
      // or that we could not sign the request (LASSO_PROFILE_ERROR_BUILDING_QUERY_FAILED).
}

// redirect user to identity provider
   // we chose the Redirect binding, so we have to generate a redirect HTTP response to the ↩
        URL returned by Lasso
printf("Location: %s\n\nRedirected to IdP\n", LASSO_PROFILE(login)->msg_url);
```

Next example shows how to receive the response from the identity provider for ID-FF 1.2.

Example 4.2 Service Provider Assertion Consumer Service URL for ID-FF 1.2

```
LassoLogin *login;
char *request_method = getenv("REQUEST_METHOD");
char *artifact_msg = NULL, *lares = NULL, *lareq = NULL;
char *name_identifier;
lassoHttpMethod method;
int rc = 0;

login = lasso_login_new(server);
if (strcmp(request_method, "GET") == 0) {
        artifact_msg = getenv("QUERY_STRING");
        method = LASSO_HTTP_METHOD_REDIRECT;
} else {
        // read submitted form; if it has a LAREQ field, put it in lareq,
        // if it has a LARES field, put it in lares
        if (lareq) {
                artifact_msg = lareq;
        } else if (lares) {
                response_msg = lares;
        } else {
                // bail out
        }
        method = LASSO_HTTP_METHOD_POST;
}

if (artifact_msg) {
        // we received an artifact response,
        // it means we did not really receive the response,
        // only a token to redeem the real response from the identity
        // provider through a SOAP resolution call
        rc = lasso_login_init_request(login, artifact_msg, method);
        if (rc != 0) {
                  ... // handle errors
                  // there is usually no error at this step, only
                  // if the IdP response is malformed
        }
        rc = lasso_login_build_request_msg(login);
        if (rc != 0) {
                  ... // handle errors
                  // as for AuthnRequest generation, it generally is caused
                  // by a bad initialization like an impossibility to load
                  // the private key.
        }
        // makes a SOAP call, soap_call is NOT a Lasso function
        soap_answer_msg = soap_call(LASSO_PROFILE(login)->msg_url,
                        LASSO_PROFILE(login)->msg_body);
        rc = lasso_login_process_response_msg(login, soap_answer_msg);
        if (rc != 0) {
                  ... // handle errors
                  // here you can know if the IdP refused the request,
        }
} else if (response_msg) {
        lasso_login_process_authn_response_msg(login, response_msg);
}

// looks up name_identifier in local file, database, whatever and gets back
// two things: identity_dump and session_dump
name_identifier = LASSO_PROFILE(login)->nameIdentifier
lasso_profile_set_identity_from_dump(LASSO_PROFILE(login), identity_dump);
lasso_profile_set_session_from_dump(LASSO_PROFILE(login), session_dump);

lasso_login_accept_sso(login);

if (lasso_profile_is_identity_dirty(LASSO_PROFILE(login))) {
        LassoIdentity *identity;
        char *identity_dump;
        identity = lasso_profile_get_identity(LASSO_PROFILE(login));
        identity_dump = lasso_identity_dump(identity);
        // record identity_dump in file, database...
}

if (lasso_profile_is_session_dirty(LASSO_PROFILE(login))) {
        LassoSession *session;
        char *session_dump;
        session = lasso_profile_get_session(LASSO_PROFILE(login));
        session_dump = lasso_session_dump(session);
        // record session_dump in file, database...
}

// redirect user anywhere
printf("Location: %s\n\nRedirected to site root\n", login->msg_url);
```

The implement an IdP you must create a single sign-on service endpoint, the needed APIs for this are lasso_login_process_authn_request_ lasso_login_validate_request_msg(), lasso_login_build_assertion(), lasso_login_build_authn_response_msg() and lasso_login_build_ar You will have to chose between lasso_login_build_authn_response_msg() and lasso_login_build_artifact_msg() depending on the requested protocol for the response by the service provider

Example 4.3 Identity provider single sign-on service

```
LassoLogin *login;
char *request_method = getenv("REQUEST_METHOD");
char *artifact_msg = NULL, *lares = NULL, *lareq = NULL;
char *name_identifier;
lassoHttpMethod method;
int rc = 0;

login = lasso_login_new(server);
if (strcmp(request_method, 'GET')) { // AuthnRequest send with the HTTP-Redirect binding
```

```
    //
    lasso_profile_set_signature_verify_hint(LASSO_PROFILE(login),
            LASSO_PROFILE_SIGNATURE_VERIFY_HINT_FORCE);
    rc = lasso_process_authn_request_msg(login, getenv("QUERY_STRING"));
    if (rc != 0) {
        // handle errors
    }


} else {
```

Functions

lasso_login_new ()

```
LassoLogin~*
lasso_login_new (LassoServer *server);
```

Creates a new LassoLogin.

Parameters

server	the LassoServer

Returns

a newly created LassoLogin object; or NULL if an error occured

lasso_login_new_from_dump ()

```
LassoLogin~*
lasso_login_new_from_dump (LassoServer *server,
                           const gchar *dump);
```

Restores the *dump* to a new LassoLogin.

Parameters

server	the LassoServer
dump	XML login dump

Returns

a newly created LassoLogin; or NULL if an error occured.

lasso_login_accept_sso ()

```
lasso_error_t
lasso_login_accept_sso (LassoLogin *login);
```

Gets the assertion of the response and adds it to the LassoSession object. Builds a federation with the 2 name identifiers of the assertion and adds it into the identity. If the session or the identity are NULL, they are created.

Parameters

login	a LassoLogin

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PROFILE_ERROR_MISSING_RESPONSE if no response is present in the login profile object; usually because no call to lasso_login_process_authn_response_msg was done;
- LASSO_PROFILE_ERROR_MISSING_ASSERTION if the response does not contain an assertion,
- LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_FOUND if the assertion does not contain a NameID element,
- LASSO_PROFILE_ERROR_MISSING_NAME_IDENTIFIER same as LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_F
- LASSO_LOGIN_ERROR_ASSERTION_REPLAY if the assertion has already been used.

lasso_login_build_artifact_msg ()

```
lasso_error_t
lasso_login_build_artifact_msg (LassoLogin *login,
                                LassoHttpMethod http_method);
```

Builds a SAML artifact. Depending of the HTTP method, the data for the sending of the artifact are stored in *msg_url* (REDIRECT) or *msg_url* , *msg_body* and *msg_relayState* (POST).

Parameters

login	a LassoLogin
http_method	the HTTP method to send the artifact (REDIRECT or POST)

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID if no remote provider ID was setup in the login profile object, it's usually done by lasso_login_process_authn_request_msg,
- LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD if the HTTP method is neither LASSO_HTTP_METHOD_REDIRECT or LASSO_HTTP_METHOD_POST (ID-FF 1.2 case) or neither LASSO_HTTP_METHOD_ARTIFACT_GET or LASSO_HTTP_M (SAML 2.0 case) for SAML 2.0),
- LASSO_PROFILE_ERROR_INVALID_PROTOCOLPROFILE if the current protocolProfile is not
- LASSO_LOGIN_PROTOCOL_PROFILE_BRWS_ART (only for ID-FF 1.2),

- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the remote provider is not known to our server object which impeach us to find a service endpoint,
- LASSO_PROFILE_ERROR_MISSING_RESPONSE if the response object is missing,
- LASSO_PROFILE_ERROR_MISSING_STATUS_CODE if the response object is missing a status code,

lasso_login_build_assertion ()

```
lasso_error_t
lasso_login_build_assertion (LassoLogin *login,
                             const char *authenticationMethod,
                             const char *authenticationInstant,
                             const char *reauthenticateOnOrAfter,
                             const char *notBefore,
                             const char *notOnOrAfter);
```

Builds an assertion and stores it in profile session. `authenticationInstant` , reauthenticateOnOrAfter, `notBefore` and `notOnOrAfter` may be NULL. If `authenticationInstant` is NULL, the current time will be used. Time values must be encoded in UTC.

Construct the authentication assertion for the response. It must be called after validating the request using lasso_login_validate_request_m The created assertion is accessed using lasso_login_get_assertion().

Parameters

login	a LassoLogin	
authenticationMethod	the authentication method	
authenticationInstant	the time at which the authentication took place	
notBefore	the earliest time instant at which the assertion is valid	
notOnOrAfter	the time instant at which the assertion has expired	

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PROFILE_ERROR_IDENTITY_NOT_FOUND if no identity object was found in the login profile object.
- LASSO_PROFILE_ERROR_MISSING_RESPONSE if no response object is present (it is normally initialized by lasso_login_process)
- LASSO_PROFILE_ERROR_FEDERATION_NOT_FOUND if a LASSO_SAML2_NAME_IDENTIFIER_FORMAT_PERSISTENT or LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENCRYPTED NameID format is asked and no corresponding federation was found in the LassoIdentity object,
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if encryption is needed and the request issuing provider is unknown (it as not been registered in the LassoServer object),
- LASSO_DS_ERROR_ENCRYPTION_FAILED if encryption is needed but it failed,

lasso_login_build_authn_request_msg ()

```
lasso_error_t
lasso_login_build_authn_request_msg (LassoLogin *login);
```

Converts profile authentication request (*request* member) into a Liberty message, either an URL in HTTP-Redirect profile or an URL and a field value in Browser-POST (form) profile.

The URL is set into the *msg_url* member and the eventual field value (LAREQ) is set into the *msg_body* member.

Parameters

login	a LassoLogin

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID if not remote provider ID was setup&160;- it usually means that lasso_login_init_request() was not called before,
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the remote provider ID is not registered in the server object,
- LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE if the SSO profile is not supported by the targeted provider,
- LASSO_PROFILE_ERROR_BUILDING_QUERY_FAILED if the building of the query part of the redirect URL or of the body of the POST content failed&160;- it only happens with the LASSO_HTTP_METHOD_REDIRECT, LASSO_HTTP_METHOD_ LASSO_HTTP_METHOD_ARTIFACT_GET and LASSO_HTTP_METHOD_ARTIFACT_POST bindings&160;-,
- LASSO_PROFILE_ERROR_UNKNOWN_PROFILE_URL if the metadata of the remote provider does not contain an url for the SSO profile,
- LASSO_PROFILE_ERROR_INVALID_REQUEST if the request object is not of the needed type, is usually means that lasso_login_init_request() was not called before,
- LASSO_PROFILE_MISSING_REQUEST if the request object is missing,
- LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD if the current setted *http_method* on the LassoLogin object is invalid.

lasso_login_build_authn_response_msg ()

```
lasso_error_t
lasso_login_build_authn_response_msg (LassoLogin *login);
```

Converts profile authentication response (*response* member) into a Liberty message.

The URL is set into the *msg_url* member and the field value (LARES) is set into the *msg_body* member.

Parameters

login	a LassoLogin

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PROFILE_ERROR_INVALID_PROTOCOLPROFILE if the current protocol profile is not
- LASSO_LOGIN_PROTOCOL_PROFILE_BRWS_POST or LASSO_LOGIN_PROTOCOL_PROFILE_BRWS_LECP,
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the remote provider ID is not registered in the server object,
- LASSO_PROFILE_ERROR_UNKNOWN_PROFILE_URL if the metadata of the remote provider does not contain an URL for the assertion consuming service,
- LASSO_PROFILE_ERROR_MISSING_SERVER the server object is needed to sign a message and it is missing,
- LASSO_DS_ERROR_PRIVATE_KEY_LOAD_FAILED the private key for signing could not be found,
- LASSO_PROFILE_ERROR_MISSING_RESPONSE if the response object is missing,
- LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE if the SSO profile is not supported by the targeted provider,
- LASSO_PROFILE_BUILDING_QUERY_FAILED if using LASSO_HTTP_METHOD_REDIRECT building of the redirect URL failed,
- LASSO_PROFILE_BUILDING_MSG_FAILED if using LASSO_HTTP_METHOD_POST, LASSO_HTTP_METHOD_SOAP or LASSO_HTTP_METHOD_PAOS and building the *msg_body* failed.

lasso_login_build_request_msg ()

```
lasso_error_t
lasso_login_build_request_msg (LassoLogin *login);
```

Produce a SOAP Artifact Resolve message. It must follows a call to lasso_login_init_request() on the artifact message. Converts artifact request into a Liberty SOAP message.

The URL is set into the *msg_url* member and the SOAP message is set into the *msg_body* member. You should POST the *msg_body* to the *msg_url* afterward.

Parameters

login	a LassoLogin

Returns

0 on success; or LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object, LASSO_PROFILE_ERR
if not remote provider ID was setup -- it usually means that lasso_login_init_request was not called before, LASSO_SERVER_ERROR_F
if the remote provider ID is not registered in the server object.

lasso_login_build_response_msg ()

```
lasso_error_t
lasso_login_build_response_msg (LassoLogin *login,
                                gchar *remote_providerID);
```

Converts profile assertion response (*response* member) into a Liberty SOAP messageresponse message.

The URL is set into the *msg_url* member and the SOAP message is set into the *msg_body* member.

Parameters

login	a LassoLogin
remote_providerID	service provider ID

Returns

0 on success; or a negative value otherwise. LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object, LASSO_PROFILE_ERROR_SESSION_NOT_FOUND if no session object was found in the login profile object -- it should be created by lasso_login_build_assertion() if you did not set it manually before calling lasso_login_build_assertion().

lasso_login_destroy ()

```
void
lasso_login_destroy (LassoLogin *login);
```

Destroys a LassoLogin object.

Deprecated : Since 2.2.1, use g_object_unref() instead.

Parameters

login	a LassoLogin

lasso_login_dump ()

```
gchar~*
lasso_login_dump (LassoLogin *login);
```

Dumps *login* content to an XML string.

Parameters

login	a LassoLogin

Returns

the dump string. It must be freed by the caller.

[transfer full]

lasso_login_get_assertion ()

```
LassoNode~*
lasso_login_get_assertion (LassoLogin *login);
```

Return the last build assertion.

Parameters

login	a LassoLogin object

Returns

a LassoNode representing the build assertion (generally a LassoSamlAssertion when using ID-FF 1.2 or a LassoSaml2Assertion when using SAML 2.0)

lasso_login_init_authn_request ()

```
lasso_error_t
lasso_login_init_authn_request (LassoLogin *login,
                                const gchar *remote_providerID,
                                LassoHttpMethod http_method);
```

Initializes a new AuthnRequest from current service provider to remote identity provider specified in *remote_providerID* (if NULL the first known identity provider is used).

For ID-FF 1.2 the default NameIDPolicy in an AuthnRequest is None, which imply that a federation must already exist on the IdP side.

For SAML 2.0 the default NameIDPolicy is the first listed in the metadatas of the current provider, or if none is specified, Transient, which ask the IdP to give a one-time federation

Parameters

login	a LassoLogin	
remote_providerID	the providerID of the identity provider (may be NULL).	*[allow-none]*
http_method	HTTP method to use for request transmission.	*[default LASSO_HTTP_METHOD_REDIRECT]*

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID if *remote_providerID* is NULL and no default remote provider could be found from the server object -- usually the first one in the order of adding to the server object --,
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the *remote_providerID* is not known to our server object.
- LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD if the HTTP method is neither LASSO_HTTP_METHOD_REDIRECT or LASSO_HTTP_METHOD_POST,
- LASSO_PROFILE_ERROR_BUILDING_REQUEST_FAILED if creation of the request object failed.

lasso_login_init_idp_initiated_authn_request ()

```
lasso_error_t
lasso_login_init_idp_initiated_authn_request
                                (LassoLogin *login,
                                const gchar *remote_providerID);
```

Generates an authentication response without matching authentication request.

The choice of NameIDFormat is the same as for lasso_login_init_authn_request() but with the target *remote_providerID* as the current provider

If *remote_providerID* is NULL, the first known provider is used.

Parameters

login	a LassoLogin.	
remote_providerID	the providerID of the remote service provider (may be NULL)	

Returns

0 on success; or a negative value otherwise. Error codes are the same as lasso_login_init_authn_request().

lasso_login_init_request ()

```
lasso_error_t
lasso_login_init_request (LassoLogin *login,
                          gchar *response_msg,
                          LassoHttpMethod response_http_method);
```

Initializes an artifact request. *response_msg* is either the query string (in redirect mode) or the form LAREQ field (in browser-post mode). It should only be used if you received an artifact message, *response_msg* must be content of the artifact field for the POST artifact binding of the query string for the REDIRECT artifact binding. You must set the *response_http_method* argument according to the way you received the artifact message.

Parameters

login	a LassoLogin	
response_msg	the authentication response received	
response_http_method	the method used to receive the authentication response	

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PARAM_ERROR_INVALID_VALUE if *response_msg* is NULL,
- LASSO_PROFILE_ERROR_INVALID_HTTP_METHOD if the HTTP method is neither LASSO_HTTP_METHOD_REDIRECT or LASSO_HTTP_METHOD_POST (in the ID-FF 1.2 case) or neither LASSO_HTTP_METHOD_ARTIFACT_GET or LASSO_HTT (in the SAML 2.0 case),
- LASSO_PROFILE_ERROR_MISSING_ARTIFACT if no artifact field was found in the query string (only possible for the LASSO_HTTP_METHOD_REDIRECT case),
- LASSO_PROFILE_ERROR_INVALID_ARTIFACT if decoding of the artifact failed -- whether because the base64 encoding is invalid or because the type code is wrong --,
- LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID if no provider ID could be found corresponding to the hash contained in the artifact.

lasso_login_must_ask_for_consent ()

```
gboolean
lasso_login_must_ask_for_consent (LassoLogin *login);
```

Evaluates if consent must be asked to the Principal to federate him.

Parameters

login	a LassoLogin

Returns

TRUE if consent must be asked

lasso_login_must_authenticate ()

```
gboolean
lasso_login_must_authenticate (LassoLogin *login);
```

Evaluates if user must be authenticated.

Parameters

login	a LassoLogin

Returns

TRUE if user must be authenticated

lasso_login_process_authn_request_msg ()

```
lasso_error_t
lasso_login_process_authn_request_msg (LassoLogin *login,
                                       const char *authn_request_msg);
```

Processes received authentication request, checks it is signed correctly, checks if requested protocol profile is supported, etc.

Parameters

login	a LassoLogin
authn_request_msg	the authentication request received

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is no a LassoLogin object,
- LASSO_PROFILE_ERROR_MISSING_REQUEST if *authn_request_msg* is NULL and no request as actually been processed or initialized &151; see lasso_login_init_idp_initiated_authn_request(),

- LASSO_PROFILE_ERROR_INVALID_MSG if the content of *authn_request_msg* cannot be parsed to as a valid lib:AuthnRequest messages for any support binding (mainly HTTP-Redirect, HTTP-Post and SOAP),
- LASSO_PROFILE_ERROR_MISSING_ISSUER if the parsed samlp2:AuthnRequest does not have a proper Issuer element,
- LASSO_PROFILE_ERROR_INVALID_REQUEST if the parsed message does not validate as a valid samlp2:AuthnRequest (SAMLv2) i.e. if there is no Issuer, or mutually exclusive attributes are used (ProtocolBinding and AssertionConsumerServiceIndex),
- LASSO_PROFILE_ERROR_INVALID_PROTOCOLPROFILE if the protocolProfile (ID-FFv1.2) or the protocolBinding (SAMLv2) is unsupported by Lasso,
- LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE if the protocolProfile (ID-FFv1.2) or the protocolBinding (SAMLv2) for the AssertionConsumer is unsupported by this provider implementation as indicated by its metadata file,
- LASSO_PROFILE_ERROR_UNKNOWN_PROVIDER, or LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the metadata for the issuer of the request are absent from the LassoServer object of this profile,
- LASSO_DS_ERROR_SIGNATURE_NOT_FOUND if no signature could be found and signature validation is forced &151; by the service provider metadata with the AuthnRequestsSigned attribute (ID-FFv1.2&SAMLv2), the attribute WantAuthnRequestsSigned in the identity provider metadata file (SAMLv2) or as advised by the lasso_profile_set_signature_verify_hint() method),
- LASSO_DS_ERROR_SIGNATURE_VERIFICATION_FAILED if the signature validation failed on a present signature,
- LASSO_DS_ERROR_INVALID_SIGNATURE if the signature was malformed and a signature was present,

lasso_login_process_authn_response_msg ()

```
lasso_error_t
lasso_login_process_authn_response_msg
                               (LassoLogin *login,
                                gchar *authn_response_msg);
```

Processes received authentication response.

Parameters

login	a LassoLogin	
authn_response_msg	the authentication response received	

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PARAM_ERROR_INVALID_VALUE if authn_response_msg is NULL,
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND, if the issuing provider of the assertion is not registered in the LassoServer object,
- LASSO_PROFILE_ERROR_MISSING_ISSUER if the parsed samlp2:AuthnRequest does not have a proper Issuer element,
- LASSO_PROFILE_ERROR_MISSING_STATUS_CODE if the reponse is missing a `StatusCode` element,
- LASSO_PROFILE_STATUS_NOT_SUCCESS_ERROR if the identity provider returned a failure response,
- LASSO_PROFILE_ERROR_REQUEST_DENIED if the identity provider returned the specific status code `RequestDenied`,

- LASSO_PROFILE_ERROR_INVALID_MSG if the message is not a LassoSamlpResponse (ID-FF 1.2) or a LassoSamlp2ResponseMs (SAML 2.0),
- LASSO_PROFILE_ERROR_UNSUPPORTED_PROFILE, if the received message format does not correspond to a binding supported by this function, the only supported binding by this function is HTTP POST,
- LASSO_PROFILE_ERROR_MISSING_SERVER the server object is needed to sign a message and it is missing,
- LASSO_PROFILE_ERROR_CANNOT_VERIFY_SIGNATURE if the validation of the signature of the message failed, a specific error code is available in `login->parent.signature_status`
- LASSO_LOGIN_ERROR_ASSERTION_DOES_NOT_MATCH_REQUEST_ID if the received response does not match the saved AuthenticationRequest ID,
- LASSO_PROFILE_ERROR_INVALID_ISSUER if the assertion issuer does not match the AuthenticationResponse issuer,
- LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_FOUND if not NameID could be found or decoded,

lasso_login_process_paos_response_msg ()

```
lasso_error_t
lasso_login_process_paos_response_msg (LassoLogin *login,
                                       gchar *msg);
```

lasso_login_process_request_msg ()

```
lasso_error_t
lasso_login_process_request_msg (LassoLogin *login,
                                 gchar *request_msg);
```

Processes received artifact request.

Parameters

login	a LassoLogin	
request_msg	the artifact request received	

Returns

0 on success; or a negative value otherwise.

lasso_login_process_response_msg ()

```
lasso_error_t
lasso_login_process_response_msg (LassoLogin *login,
                                  gchar *response_msg);
```

Processes received assertion response.

Parameters

login	a LassoLogin	
response_msg	the assertion response received	

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_PARAM_ERROR_INVALID_VALUE if response_msg is NULL,
- LASSO_PROFILE_ERROR_INVALID_MSG if the message is not a LassoSamlpResponse (ID-FF 1.2) or a LassoSamlp2ResponseMs (SAML 2.0),
- LASSO_PROFILE_ERROR_RESPONSE_DOES_NOT_MATCH_REQUEST if the response does not refer to the request or if the response refer to an unknown request and strict-checking is activated ,
- LASSO_LOGIN_ERROR_REQUEST_DENIED the identity provided returned a failure status of "RequestDenied"
- LASSO_LOGIN_ERROR_FEDERATION_NOT_FOUND if creation of a new federation was not allowed and none existed,
- LASSO_LOGIN_ERROR_UNKNOWN_PRINCIPAL if authentication failed or/and if the user cancelled the authentication,
- LASSO_LOGIN_ERROR_STATUS_NOT_SUCCESS, if the response status is a failure but we have no more precise error code to report it, you must look at the second level status in the response,
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND, if the issuing provider of the assertion is unknown,
- LASSO_PROFILE_ERROR_INVALID_ISSUER the issuer of the assertion received, is not the expected one
- LASSO_PROFILE_ERROR_NAME_IDENTIFIER_NOT_FOUND no statement was fournd, or none statement contains a subject with a name identifier,
- LASSO_PROFILE_ERROR_MISSING_STATUS_CODE if the reponse is missing a `StatusCode` element,
- LASSO_PROFILE_ERROR_MISSING_ASSERTION if the message does not contain any assertion.

lasso_login_validate_request_msg ()

```
lasso_error_t
lasso_login_validate_request_msg (LassoLogin *login,
                                  gboolean authentication_result,
                                  gboolean is_consent_obtained);
```

Initializes a response to the authentication request received.

Parameters

login	a LassoLogin	
authentication_result	whether user has authenticated succesfully	
is_consent_obtained	whether user consent has been obtained	

Returns

0 on success; or

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object,
- LASSO_LOGIN_ERROR_REQUEST_DENIED if *authentication_result* if FALSE,
- LASSO_LOGIN_ERROR_INVALID_SIGNATURE if signature validation of the request failed,

- LASSO_LOGIN_ERROR_UNSIGNED_AUTHN_REQUEST if no signature was present on the request,
- LASSO_LOGIN_ERROR_FEDERATION_NOT_FOUND if federation policy is LASSO_LIB_NAMEID_POLICY_TYPE_NONE and no federation was found in the LassoIdentity object (ID-FF 1.2 case)
- LASSO_LOGIN_ERROR_INVALID_NAMEIDPOLICY if request policy is not one of LASSO_LIB_NAMEID_POLICY_TYPE_FE or LASSO_LIB_NAMEID_POLICY_TYPE_ANY (ID-FF 1.2 case) or if no NameID policy was defined or the AllowCreate request flag is FALSE (SAML 2.0 case),
- LASSO_LOGIN_ERROR_CONSENT_NOT_OBTAINED if *is_consent_obtained* is FALSE and conssent was necessary (for example if the request does not communicate that consent was already obtained from the user),
- LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the requesting provider is unknown,

Types and Values

struct LassoLogin

```
struct LassoLogin {
  LassoProfile parent;

  LassoLoginProtocolProfile protocolProfile;
  gchar *assertionArtifact;
};
```

Single sign-on profile for the current transaction; possibly an assertionArtifact to be used by the service provider in its "assertionConsumerServiceURL" and the assertion created or received for the principal.

Members

LassoProfile *parent*;		
LassoLoginProtocolProfile *protocolProfile*;		the kind of binding used for this authentication request.

gchar **assertionArtifact*;	a string representing the artifact received through an artifact resolution. request

4.2 LassoLogout

LassoLogout — Single Logout Profile

Functions

LassoLogout *	lasso_logout_new ()
LassoLogout *	lasso_logout_new_from_dump ()
lasso_error_t	lasso_logout_build_request_msg ()
lasso_error_t	lasso_logout_build_response_msg ()
void	lasso_logout_destroy ()
gchar *	lasso_logout_dump ()
gchar *	lasso_logout_get_next_providerID ()
lasso_error_t	lasso_logout_init_request ()
lasso_error_t	lasso_logout_process_request_msg ()
lasso_error_t	lasso_logout_process_response_msg ()
lasso_error_t	lasso_logout_reset_providerID_index ()
lasso_error_t	lasso_logout_validate_request ()

Types and Values

struct	LassoLogout

Description

This profile Send logout notifications between providers. Any receiving provider must retransmit the notification to any other providers with which it shares the current identity by any means supported by the two, that is any provider federated with the current provider. There can be partial failures if no binding can be found to notify a federating partner or if a partner fails to respond.

It is generally advised to apply the local logout transaction before sending a logout request to a partner. In short:

- an identity provider receiving a logout request should kill the local session before sending logout request to other service provider and proxyied identity providers.
- a service provider intitiating a logout request must first kill its local session, then proceeds with the logout exchange with its identity provider

The following examples must not be used 'as-is' they lack most of the error checking code that is needed for a secured and robust program, but they give an idea of how to use the API

Example 4.4 Service Provider Initiated Logout

```
LassoLogout *logout;
char *session_dump; // must contain the session dump
                    // for the current user
int rc; // hold return codes
char *soap_response;

LassoHttpMethod method; // method to use, LASSO_HTTP_METHOD_REDIRECT,
                        // LASSO_HTTP_METHOD_POST or LASSO_HTTP_METHOD_SOAP,
                        // other methods are rarely supported

logout = lasso_logout_new(server);
lasso_profile_set_session_from_dump(&logout->parent, session_dump);
// the second argument can be NULL, lasso_logout_init_request() will automatically choose ←
    the
// identity provider from the first assertion int the session
rc = lasso_logout_init_request(logout, "http://identity-provider-id/",
                method);
if (rc != 0) {
  ... // handle errors, most of them are related to bad initialization
      // or unsupported binding
}
rc = lasso_logout_build_request_msg(logout);
if (rc != 0) {
  ... // handle errors, most of them are related to bad initialization
      // or impossibility to build the query string (missing private keys for signing)
}

// now send the request
switch (method) {
    case LASSO_HTTP_METHOD_REDIRECT:
        // LASSO_PROFILE(logout)->msg_url contains the URL where the
        // User Agent must be redirected
        ...
        // save the session and logout object, and store them attached to the RequestID of ←
            the
        // request, you will need them for handling the response
        session_dump = lasso_node_dump((LassoNode*)logout->parent.session);
        logout_dump = lasso_node_dump((LassoNode*)logout);
        break;
    case LASSO_HTTP_METHOD_POST:
        // you must build a form with a field name SAMLRequest (SAML 2.0) or LAREQ (ID-FF ←
            1.2)
        // with the content of LASSO_PROFILE(logout)->msg_body
        // posting to the address LASSO_PROFILE(logout)->msg_url
        ...
        // save the session and logout object, and store them attached to the RequestID of ←
            the
        // request, you will need them for handling the response
        session_dump = lasso_node_dump((LassoNode*)logout->parent.session);
        logout_dump = lasso_node_dump((LassoNode*)logout);
        break;
    case LASSO_HTTP_SOAP:
        // makes a SOAP call, soap_call is NOT a Lasso function
        soap_response = soap_call(login->parent.msg_url, login->parent.msg_body);
        rc = lasso_logout_process_response_msg(logout, soap_response);
        if (rc != 0) {
            // handle errors, important ones are LASSO_LOGOUT_ERROR_UNSUPPORTED_PROFILE  ↩
                meaning
            // that one other service provider of the current session cannot be contacted  ↩
                by the
            // identity provider with the current binding, for example it only accept  ↩
                REDIRECT
            (asynchronous-binding) or
            // POST an we are using SOAP (synchronous-binding).
            ...
        }
        // everything is ok save the session
        session_dump = lasso_node_dump(logout->parent.session);
        // nothing to save because you killed the local session already
        break;
    default:
        // other binding neither are frequent or largely supported
        // so report an error
        break;
    }
```

The next example show the endpoint for handling response to request with asynchronous binding (POST and Redirect).

Example 4.5 Service Provider Logout Request Endpoint

```
LassoLogout *logout;
char *request_method = getenv("REQUEST_METHOD");

logout = lasso_logout_new(server);

if (strcmp(request_method, "GET") == 0) {
    char query_string = getenv("QUERY_STRING");
    rc = lasso_logout_process_response_msg(logout, query_string);
} elif (strcmp(request_method, "POST") == 0) {
    char *message;
    // message should contain the content of LARES or SAMLResponse fied, depending if this  ↩
        is an
    // ID-FF 1.2 or SAML 2.0 service.
    rc = lasso_logout_process_response_msg(logout, message);
}
if (rc != 0) {
    // handle errors, as we are already unlogged, those must go to a log file or audit  ↩
        trail,
    // because at this time the user do not care anymore. A report about a failure to  ↩
        logout to
    // the IdP can be eventually shown.
    ...
}
```

The next snippet show how to implement a logout endpoint, to receive a logout request and respond.

Example 4.6 Service Provider Logout Request Endpoint

```
LassoLogout *logout;
char *session_dump;
char *request_method = getenv("REQUEST_METHOD");
int rc;
int method;

logout = lasso_logout_new(server);
// server must be previously initialized, it can be kept around
// and used for many transaction, it is never modified by any profile
if (strcmp(request_method. "GET") == 0) {
    method = LASSO_HTTP_METHOD_REDIRECT;
    char query_string = getenv("QUERY_STRING");
    rc = lasso_logout_process_request_msg(logout, query_string);
    if (rc != 0) {
        // handle errors
        ...
    }
} else if (strcmp(request_method, "POST") == 0) {
    char *message;
    // read submitted content if this is a form, put LAREQ or SAMLRequest field into ↩
       message and
    set method to LASSO_HTTP_METHOD_POST
    // if content type is application/xml then put the full body of the POST inside message ↩
         and
    // set method to LASSO_HTTP_METHOD_SOAP
    rc = lasso_logout_process_request_msg(logout, message);
    if (rc != 0) {
        // handle errors
        ...
    }
}
protocolProfile = lasso_provider_get_protocol_conformance(LASSO_PROVIDER(server));
if (protocolProfile == LASSO_LIBERTY_1_2) {
    char *session_index;
    LassoSamlNameIdentifier *name_id;
    LibLogoutRequest *logout_request;

    logout_request = LIB_LOGOUT_REQUEST(LASSO_PROFILE(logout)->request);
    session_index = logout_request->SessionIndex;
    name_id = logout_request->NameIdentifier;
    // lookup the session dump using session_index and name_id
} else if (protocolProfile == LASSO_SAML_2_0) {
    char *session_index;
    LassoSaml2NameID *name_id;
    LassoSamlp2LogoutRequest *logout_request;

    logout_request = LASSO_SAMLP2_LOGOUT_REQUEST(LASSO_PROFILE(logout)->request);
    session_index = logout_request->SessionIndex;
    name_id = logout_request->NameID;
    // lookup the session dump using session_index and name_id
}
lasso_profile_set_session_from_dump(LASSO_PROFILE(logout), session_dump);
// you can check other property of the request here if you want
//
if (request is accepted) {
    rc = lasso_logout_validate_request(logout);
    if (rc != 0) {
        // handle errors..
        ...
    } else {
    .... // kill the local session
         // if local server is an identity provider, then traverse the session using
         // lasso_logout_get_next_providerID() and send logout request to all logged
         // service providers.
    }
}
// if lasso_logout_validate_request() was not called this will automatically create a  ↩
    Failure
// response.
rc = lasso_logout_build_response_msg(logout);
if (rc != 0) {
    // handle errors..
    ...
}
// the response is produced with the same binding as the request
// see the previous request example for how to send the response
// the only change is for SOAP, you just need to print the msg_body as page content with a
// Content-type of application/xml.
```

Functions

lasso_logout_new ()

```
LassoLogout~*
lasso_logout_new (LassoServer *server);
```

Creates a new LassoLogout.

Parameters

server	the LassoServer

Returns

a newly created LassoLogout object; or NULL if an error occured

lasso_logout_new_from_dump ()

```
LassoLogout~*
lasso_logout_new_from_dump (LassoServer *server,
                            const gchar *dump);
```

Restores the *dump* to a new LassoLogout.

Parameters

server	the LassoServer
dump	XML logout dump

Returns

a newly created LassoLogout; or NULL if an error occured

lasso_logout_build_request_msg ()

```
lasso_error_t
lasso_logout_build_request_msg (LassoLogout *logout);
```

Builds the logout request message.

It gets the HTTP method retrieved to send the request and:

- if it is a SOAP method, then it builds the logout request SOAP message, sets the msg_body attribute, gets the single logout service url and sets *msg_url* in the logout object.
- if it is a HTTP-Redirect method, then it builds the logout request QUERY message, builds the logout request url, sets *msg_url* in the logout request url, sets *msg_body* to NULL.

If private key and certificate are set in server object it will also signs the message (either with X509 if SOAP or with a simple signature for query strings).

Parameters

logout	a LassoLogout

Returns

0 on success; or a negative value otherwise.

lasso_logout_build_response_msg ()

```
lasso_error_t
lasso_logout_build_response_msg (LassoLogout *logout);
```

Builds the logout response message.

It gets the request message method and:

- if it is a SOAP method, then it builds the logout response SOAP message, sets the msg_body attribute, gets the single logout service return url and sets *msg_url* in the logout object.
- if it is a HTTP-Redirect method, then it builds the logout response QUERY message, builds the logout response url, sets *msg_url* with the logout response url, sets *msg_body* to NULL

If private key and certificate are set in server object it will also signs the message (either with X509 if SOAP or with a simple signature for query strings).

Parameters

logout	a LassoLogout

Returns

0 on success; or a negative value otherwise.

lasso_logout_destroy ()

```
void
```

```
lasso_logout_destroy (LassoLogout *logout);
```

Destroys a logout object.

Parameters

logout	a LassoLogout

lasso_logout_dump ()

```
gchar~*
lasso_logout_dump (LassoLogout *logout);
```

Dumps *logout* content to an XML string.

Parameters

logout	a LassoLogout

Returns

the dump string. It must be freed by the caller.

[transfer full]

lasso_logout_get_next_providerID ()

```
gchar~*
lasso_logout_get_next_providerID (LassoLogout *logout);
```

Returns the provider id from providerID_index in list of providerIDs in principal session with the exception of initial service provider ID.

Parameters

logout	a LassoLogout

Returns

a newly allocated string or NULL.

[transfer full]

lasso_logout_init_request ()

```
lasso_error_t
lasso_logout_init_request (LassoLogout *logout,
                           gchar *remote_providerID,
                           LassoHttpMethod request_method);
```

Initializes a new SLO request.

Parameters

logout	a LassoLogout	
remote_providerID	the providerID of the identity provider. If NULL the first identity provider is used.	
request_method	if set, then it get the protocol profile in metadata corresponding of this HTTP request method.	

Returns

0 on success; or a negative value otherwise.

lasso_logout_process_request_msg ()

```
lasso_error_t
lasso_logout_process_request_msg (LassoLogout *logout,
                                  gchar *request_msg);
```

Processes a SLO LogoutRequest message. Rebuilds a request object from the message and optionally verifies its signature.

Parameters

logout	a LassoLogout	
request_msg	the logout request message	

Returns

0 on success; or a negative value otherwise.

lasso_logout_process_response_msg ()

```
lasso_error_t
lasso_logout_process_response_msg (LassoLogout *logout,
                                   gchar *response_msg);
```

Parses the response message and builds the response object.

Checks the status code value and if it is not success, then if the local provider is a Service Provider and response method is SOAP, then builds a new logout request message for HTTP Redirect / GET method and returns the error code LASSO_LOGOUT_ERROR_UNS

If it is a SOAP method or, IDP type and http method is Redirect/GET, then removes assertion.

If local server is an Identity Provider and if there is no more assertion (Identity Provider has logged out every Service Providers), then restores the initial response.

Parameters

logout	a LassoLogout	
response_msg	the response message	

Returns

0 on success; or a negative value otherwise.

lasso_logout_reset_providerID_index ()

```
lasso_error_t
lasso_logout_reset_providerID_index (LassoLogout *logout);
```

Reset the providerID_index attribute (set to 0).

Parameters

logout	a LassoLogout

Returns

0 on success; or a negative value otherwise.

lasso_logout_validate_request ()

```
lasso_error_t
lasso_logout_validate_request (LassoLogout *logout);
```

- Sets the remote provider id
- Sets a logout response with status code value to success.
- Checks current signature status, if verification failed, stop processing and set the status code value to failure.
- Verifies federation and authentication.
- If the request http method is a SOAP method, then verifies every other Service Providers supports SOAP method : if not, then sets status code value to UnsupportedProfile and returns a code error with LASSO_LOGOUT_ERROR_UNSUPPORTED_PROFILE.
- Every tests are ok, then removes assertion.
- If local server is an Identity Provider and if there is more than one Service Provider (except the initial Service Provider), then saves the initial request, response and remote provider id.

Parameters

logout	a LassoLogout

Returns

0 on success; or LASSO_PROFILE_ERROR_MISSING_REQUEST if no request has been found -- usually means that lasso_logout_pro was not called, LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the requesting provider is not known to the server object, LASSO_PROFILE_ERROR_BUILDING_RESPONSE_FAILED if creation of the response object failed, LASSO_PROFILE_ERR if the request do not contain a NameID element, LASSO_PROFILE_ERROR_SESSION_NOT_FOUND if the logout profile object do not contain a session object, LASSO_PROFILE_ERROR_MISSING_ASSERTION if no assertion from the requesting provider was found, LASSO_PROFILE_ERROR_IDENTITY_NOT_FOUND if the logout profile object do not contain an identity object, LASSO_PROFILE_ERROR_FEDERATION_NOT_FOUND if no federation for the requesting provider was found, LASSO_LOGOUT_ERROR_UNSUPPORTED_PROFILE if the requested HTTP method is not supported by all the remote

provider of the current session.

Types and Values

struct LassoLogout

```
struct LassoLogout {
  LassoProfile parent;
};
```

4.3 LassoDefederation

LassoDefederation — Federation Termination Notification Profile (ID-FF)

Functions

LassoDefederation *	lasso_defederation_new ()
lasso_error_t	lasso_defederation_build_notification_msg ()
void	lasso_defederation_destroy ()
lasso_error_t	lasso_defederation_init_notification ()
lasso_error_t	lasso_defederation_process_notification_msg ()
lasso_error_t	lasso_defederation_validate_notification ()

Types and Values

struct	LassoDefederation

Description

The Federation Termination Notification Profiles serves to suppress federations between identity providers and services providers. It can be initiated by any of the partners using Redirect or SOAP binding.

Functions

lasso_defederation_new ()

```
LassoDefederation~*
lasso_defederation_new (LassoServer *server);
```

Creates a new LassoDefederation.

Parameters

server	the LassoServer

Returns

a newly created LassoDefederation object; or NULL if an error occured

lasso_defederation_build_notification_msg ()

```
lasso_error_t
lasso_defederation_build_notification_msg
                               (LassoDefederation *defederation);
```

Builds the federation termination notification message.

It gets the federation termination notification protocol profile and:

- if it is a SOAP method, then it builds the federation termination notification SOAP message, optionally signs the notification node, sets *msg_body*, gets the SoapEndpoint url and sets *msg_url* of the federation termination object.
- if it is a HTTP-Redirect method, then it builds the federation termination notification QUERY message (optionally signs the notification message), builds the federation termination notification url with federation termination service url, sets *msg_url* in the federation termination object, sets *msg_body* to NULL.

Parameters

defederation	a LassoDefederation	

Returns

0 on success; or a negative value otherwise.

lasso_defederation_destroy ()

```
void
lasso_defederation_destroy (LassoDefederation *defederation);
```

Destroys a LassoDefederation object.

Parameters

defederation	a LassoDefederation	

lasso_defederation_init_notification ()

```
lasso_error_t
lasso_defederation_init_notification (LassoDefederation *defederation,
                                      gchar *remote_providerID,
                                      LassoHttpMethod http_method);
```

Sets a new federation termination notification to the remote provider id with the provider id of the requester (from the server object) and the name identifier of the federated principal.

Parameters

defederation	a LassoDefederation	
remote_providerID	the provider id of the federation termination notified provider.	
http_method	the HTTP method to send the message.	

Returns

0 on success; or a negative value otherwise.

lasso_defederation_process_notification_msg ()

```
lasso_error_t
lasso_defederation_process_notification_msg
                               (LassoDefederation *defederation,
                                gchar *notification_msg);
```

Processes a lib:FederationTerminationNotification message. Rebuilds a request object from the message and optionally verifies its signature.

Set the msg_nameIdentifier attribute with the NameIdentifier content of the notification object and optionally set the msg_relayState attribute with the RelayState content of the notification object.

Parameters

defederation	the federation termination object	
notification_msg	the federation termination notification message	

Returns

0 on success; or a negative value otherwise.

lasso_defederation_validate_notification ()

```
lasso_error_t
lasso_defederation_validate_notification
                               (LassoDefederation *defederation);
```

Checks notification with regards to message status and principal federations; update them accordingly.

Parameters

defederation	a LassoDefederation	

Returns

0 on success; or a negative value otherwise.

Types and Values

struct LassoDefederation

```
struct LassoDefederation {
  LassoProfile parent;
};
```

4.4 LassoNameRegistration

LassoNameRegistration — Name Registration Profile (ID-FF)

Functions

LassoNameRegistration *	lasso_name_registration_new ()
LassoNameRegistration *	lasso_name_registration_new_from_dump ()
lasso_error_t	lasso_name_registration_build_request_msg ()
lasso_error_t	lasso_name_registration_build_response_msg ()
void	lasso_name_registration_destroy ()
gchar *	lasso_name_registration_dump ()
lasso_error_t	lasso_name_registration_init_request ()
lasso_error_t	lasso_name_registration_process_request_msg ()
lasso_error_t	lasso_name_registration_process_response_msg ()
lasso_error_t	lasso_name_registration_validate_request ()

Types and Values

struct	LassoNameRegistration

Description

Functions

lasso_name_registration_new ()

```
LassoNameRegistration~*
lasso_name_registration_new (LassoServer *server);
```

Creates a new LassoNameRegistration.

Parameters

server	the LassoServer

Returns

a newly created LassoNameRegistration object; or NULL if an error occured

lasso_name_registration_new_from_dump ()

```
LassoNameRegistration~*
lasso_name_registration_new_from_dump (LassoServer *server,
                                       const char *dump);
```

Restores the *dump* to a new LassoNameRegistration.

Parameters

server	the LassoServer
dump	XML logout dump

Returns

a newly created LassoNameRegistration; or NULL if an error occured

lasso_name_registration_build_request_msg ()

```
lasso_error_t
lasso_name_registration_build_request_msg
                                (LassoNameRegistration *name_registration);
```

Builds a register name identifier request message.

It gets the register name identifier protocol profile and:

- if it is a SOAP method, then it builds the register name identifier request SOAP message, optionally signs his node, sets *msg_body*, gets the SoapEndpoint url and sets *msg_url*.
- if it is a HTTP-Redirect method, then it builds the register name identifier request QUERY message (optionally signs the request message), builds the request url with register name identifier url with register name identifier service url, sets *msg_url* in the register name identifier object, sets *msg_body* to NULL.

Parameters

name_registration	a LassoNameRegistration

Returns

0 on success; or a negative value otherwise.

lasso_name_registration_build_response_msg ()

```
lasso_error_t
lasso_name_registration_build_response_msg
                                (LassoNameRegistration *name_registration);
```

Builds the register name idendifier response message.

It gets the request message method and:

- if it is a SOAP method, then it builds the response SOAP message, sets the msg_body attribute, gets the register name identifier service return url and sets *msg_url* of the object.
- if it is a HTTP-Redirect method, then it builds the response QUERY message, builds the response url, sets *msg_url* with the response url and sets the msg_body with NULL

If private key and certificate are set in server object it will also signs the message (either with X509 if SOAP or with a simple signature for query strings).

Parameters

name_registration	a LassoNameRegistration

Returns

0 on success; or a negative value otherwise.

lasso_name_registration_destroy ()

```
void
lasso_name_registration_destroy (LassoNameRegistration *name_registration);
```

Destroys a LassoNameRegistration object.

Parameters

name_registration	a LassoNameRegistration

lasso_name_registration_dump ()

```
gchar~*
lasso_name_registration_dump (LassoNameRegistration *name_registration);
```

Dumps `name_registration` content to an XML string.

Parameters

name_registration	a LassoNameRegistration

Returns

the dump string. It must be freed by the caller.

[transfer full]

lasso_name_registration_init_request ()

```
lasso_error_t
lasso_name_registration_init_request (LassoNameRegistration *name_registration,
                                      char *remote_providerID,
                                      LassoHttpMethod http_method);
```

Initializes a new lib:RegisterNameIdentifierRequest request; it sets `name_registration->nameIdentifier` to the new name identifier and `name_registration->oldNameIdentifier` to the old one.

Parameters

name_registration	a LassoNameRegistration
remote_providerID	the providerID of the identity provider.
http_method	if set, then it get the protocol profile in metadata corresponding of this HTTP request method.

Returns

0 on success; or a negative value otherwise.

lasso_name_registration_process_request_msg ()

```
lasso_error_t
lasso_name_registration_process_request_msg
                               (LassoNameRegistration *name_registration,
                                gchar *request_msg);
```

Processes a lib:RegisterNameIdentifierRequest message. Rebuilds a request object from the message and optionally verifies its signature. Sets profile->nameIdentifier to local name identifier. If it changed (when this is IdP-initiated and there was no previously defined local name identifier) profile->nameIdentifier will be the new one and profile->oldNameIdentiifer the old one.

Parameters

name_registration	a LassoNameRegistration	
request_msg	the register name identifier request message	

Returns

0 on success; or a negative value otherwise.

lasso_name_registration_process_response_msg ()

```
lasso_error_t
lasso_name_registration_process_response_msg
                               (LassoNameRegistration *name_registration,
                                gchar *response_msg);
```

Processes a lib:RegisterNameIdentifierResponse message. Rebuilds a response object from the message and optionally verifies its signature.

If the response depicts Success it will also update Principal federation.

Parameters

name_registration	a LassoNameRegistration	
response_msg	the register name identifier response message	

Returns

0 on success; or a negative value otherwise.

lasso_name_registration_validate_request ()

```
lasso_error_t
lasso_name_registration_validate_request
                               (LassoNameRegistration *name_registration);
```

Checks profile request with regards to message status and principal federations, update them accordingly and prepares a lib:RegisterName
accordingly.

Parameters

name_registration	a LassoNameRegistration

Returns

0 on success; or a negative value otherwise.

Types and Values

struct LassoNameRegistration

```
struct LassoNameRegistration {
  LassoProfile parent;

  LassoSamlNameIdentifier *oldNameIdentifier;
};
```

4.5 LassoNameIdentifierMapping

LassoNameIdentifierMapping — Liberty Enabled Client and Proxy Profile (ID-FF)

Functions

LassoNameIdentifierMapping *	lasso_name_identifier_mapping_new ()
lasso_error_t	lasso_name_identifier_mapping_build_request_msg ()
lasso_error_t	lasso_name_identifier_mapping_build_response_msg ()
void	lasso_name_identifier_mapping_destroy ()
lasso_error_t	lasso_name_identifier_mapping_init_request ()
lasso_error_t	lasso_name_identifier_mapping_process_request_msg ()
lasso_error_t	lasso_name_identifier_mapping_process_response_msg ()
lasso_error_t	lasso_name_identifier_mapping_validate_request ()

Types and Values

struct	LassoNameIdentifierMapping

Description

Functions

lasso_name_identifier_mapping_new ()

```
LassoNameIdentifierMapping~*
lasso_name_identifier_mapping_new (LassoServer *server);
```

Creates a new LassoNameIdentifierMapping.

Parameters

server	the LassoServer

Returns

a newly created LassoNameIdentifierMapping object; or NULL if an error occured

lasso_name_identifier_mapping_build_request_msg ()

```
lasso_error_t
lasso_name_identifier_mapping_build_request_msg
                               (LassoNameIdentifierMapping *mapping);
```

Builds a name identifier mapping request message.

- If it is a SOAP method, then it builds the request as a SOAP message, optionally signs his node, sets *msg_body* with that message and sets *msg_url* with the SOAP Endpoint URL
- If it is a HTTP-Redirect method, then it builds the request as a query string message, optionally signs it and sets *msg_url* to that URL.

Parameters

mapping	a LassoNameIdentifierMapping

Returns

0 on success; or a negative value otherwise.

lasso_name_identifier_mapping_build_response_msg ()

```
lasso_error_t
lasso_name_identifier_mapping_build_response_msg
                               (LassoNameIdentifierMapping *mapping);
```

Builds a name identifier mapping response message.

- If it is a SOAP method, then it builds the response as a SOAP message, optionally signs his node, sets *msg_body* with that message and sets *msg_url* with the register name identifier service return URL.
- If it is a HTTP-Redirect method, then it builds the response as a query string message, optionally signs it and sets *msg_url* to that URL.

If private key and certificate are set in server object it will also signs the message (either with X509 if SOAP or with a simple signature for query strings).

Parameters

mapping	a LassoNameIdentifierMapping

Returns

0 on success; or a negative value otherwise.

lasso_name_identifier_mapping_destroy ()

```
void
lasso_name_identifier_mapping_destroy (LassoNameIdentifierMapping *mapping);
```

Destroys a LassoNameIdentifierMapping object.

Parameters

mapping	a LassoNameIdentifierMapping

lasso_name_identifier_mapping_init_request ()

```
lasso_error_t
lasso_name_identifier_mapping_init_request
                               (LassoNameIdentifierMapping *mapping,
                                gchar *targetNamespace,
                                gchar *remote_providerID);
```

Initializes a new lib:NameIdentifierMappingRequest request.

Parameters

mapping	a LassoNameIdentifierMapping
targetNamespace	the request targetNamespace
remote_providerID	the providerID of the identity provider.

Returns

0 on success; or a negative value otherwise.

lasso_name_identifier_mapping_process_request_msg ()

```
lasso_error_t
lasso_name_identifier_mapping_process_request_msg
                               (LassoNameIdentifierMapping *mapping,
                                gchar *request_msg);
```

Processes a lib:NameIdentifierMappingRequest message. Rebuilds a request object from the message and optionally verifies its signature.

Parameters

mapping	a LassoNameIdentifierMapping
request_msg	the name identifier mapping request message

Returns

0 on success; or a negative value otherwise.

lasso_name_identifier_mapping_process_response_msg ()

```
lasso_error_t
lasso_name_identifier_mapping_process_response_msg
                             (LassoNameIdentifierMapping *mapping,
                              gchar *response_msg);
```

Processes a lib:NameIdentifierMappingResponse message. Rebuilds a response object from the message and optionally verifies its signature.

If the response depicts Success it will also sets `targetNameIdentifier` .

Parameters

mapping	a LassoNameIdentifierMapping
response_msg	the name identifier mapping response message

Returns

0 on success; or a negative value otherwise.

lasso_name_identifier_mapping_validate_request ()

```
lasso_error_t
lasso_name_identifier_mapping_validate_request
                             (LassoNameIdentifierMapping *mapping);
```

Checks profile request with regards to message status and principal federations, update them accordingly and prepares a lib:NameIdentifie accordingly.

Parameters

mapping	a LassoNameIdentifierMapping

Returns

0 on success; or a negative value otherwise.

Types and Values

struct LassoNameIdentifierMapping

```
struct LassoNameIdentifierMapping {
  LassoProfile parent;

  gchar *targetNameIdentifier;
};
```

4.6 LassoLecp

LassoLecp — Liberty Enabled Client and Proxy Profile (ID-FF)

Functions

LassoLecp *	lasso_lecp_new ()
lasso_error_t	lasso_lecp_build_authn_request_envelope_msg ()
lasso_error_t	lasso_lecp_build_authn_request_msg ()
lasso_error_t	lasso_lecp_build_authn_response_msg ()
lasso_error_t	lasso_lecp_build_authn_response_envelope_msg ()
void	lasso_lecp_destroy ()
lasso_error_t	lasso_lecp_init_authn_request ()
lasso_error_t	lasso_lecp_process_authn_request_msg ()
lasso_error_t	lasso_lecp_process_authn_request_envelope_msg ()
lasso_error_t	lasso_lecp_process_authn_response_envelope_msg ()

Types and Values

struct	LassoLecp

Description

Functions

lasso_lecp_new ()

```
LassoLecp~*
lasso_lecp_new (LassoServer *server);
```

Creates a new LassoLecp.

Parameters

server	the LassoServer

Returns

a newly created LassoLecp object; or NULL if an error occured

lasso_lecp_build_authn_request_envelope_msg ()

```
lasso_error_t
lasso_lecp_build_authn_request_envelope_msg
                               (LassoLecp *lecp);
```

Builds an enveloped authentication request message. Sets *msg_body* to that message.

Parameters

lecp	a LassoLecp

Returns

0 on success; or a negative value otherwise.

lasso_lecp_build_authn_request_msg ()

```
lasso_error_t
lasso_lecp_build_authn_request_msg (LassoLecp *lecp);
```

Builds an authentication request. The data for the sending of the request are stored in *msg_url* and *msg_body* (SOAP POST).

Parameters

lecp	a LassoLecp

Returns

0 on success; or a negative value otherwise.

lasso_lecp_build_authn_response_msg ()

```
lasso_error_t
lasso_lecp_build_authn_response_msg (LassoLecp *lecp);
```

Builds the lecp authentication response message (base64). Sets *msg_body* to that message.

Parameters

lecp	a LassoLecp

Returns

0 on success; or a negative value otherwise.

lasso_lecp_build_authn_response_envelope_msg ()

```
lasso_error_t
lasso_lecp_build_authn_response_envelope_msg
                               (LassoLecp *lecp);
```

Builds the enveloped LECP authentication response message (SOAP message). Sets *msg_body* to that message.

Parameters

lecp	a LassoLecp

Returns

0 on success; or a negative value otherwise.

lasso_lecp_destroy ()

```
void
lasso_lecp_destroy (LassoLecp *lecp);
```

Destroys a LassoLecp object

Parameters

lecp	a LassoLecp

lasso_lecp_init_authn_request ()

```
lasso_error_t
lasso_lecp_init_authn_request (LassoLecp *lecp,
                               const char *remote_providerID);
```

Initializes a new lib:AuthnRequest.

Parameters

lecp	a LassoLecp
remote_providerID	the providerID of the identity provider. When NULL, the first known identity provider is used.

Returns

0 on success; or a negative value otherwise.

lasso_lecp_process_authn_request_msg ()

```
lasso_error_t
lasso_lecp_process_authn_request_msg (LassoLecp *lecp,
                                      const char *authn_request_msg);
```

Processes received authentication request, checks it is signed correctly, checks if requested protocol profile is supported, etc.

Parameters

lecp	a LassoLecp	
authn_request_msg	the authentication request received	

Returns

0 on success; or a negative value otherwise.

lasso_lecp_process_authn_request_envelope_msg ()

```
lasso_error_t
lasso_lecp_process_authn_request_envelope_msg
                                (LassoLecp *lecp,
                                 const char *request_msg);
```

Processes received enveloped authentication request, extracts the authentication request out of it.

Parameters

lecp	a LassoLecp	
request_msg	the enveloped authentication request received	

Returns

0 on success; or a negative value otherwise.

lasso_lecp_process_authn_response_envelope_msg ()

```
lasso_error_t
lasso_lecp_process_authn_response_envelope_msg
                                (LassoLecp *lecp,
                                 const char *response_msg);
```

Processes received enveloped authentication response, extracts the authentication response out of it and stores it in *response* .

Parameters

lecp	a LassoLecp	
response_msg	the enveloped authentication response received	

Returns

0 on success; or a negative value otherwise.

Types and Values

struct LassoLecp

```
struct LassoLecp {
  LassoLogin parent;

  LassoLibAuthnRequestEnvelope *authnRequestEnvelope;
  LassoLibAuthnResponseEnvelope *authnResponseEnvelope;
  char *assertionConsumerServiceURL;
};
```

Chapter 5

Objects from ID-FF 1.2 schemas

5.1 Strings for ID-FF 1.2

Strings for ID-FF 1.2 —

Types and Values

#define	LASSO_LIB_HREF
#define	LASSO_LIB_PREFIX
#define	LASSO_METADATA_HREF
#define	LASSO_METADATA_PREFIX
#define	LASSO_SAML_ASSERTION_HREF
#define	LASSO_SAML_ASSERTION_PREFIX
#define	LASSO_SAML_PROTOCOL_HREF
#define	LASSO_SAML_PROTOCOL_PREFIX
#define	LASSO_PAOS_HREF
#define	LASSO_PAOS_PREFIX
#define	LASSO_ECP_HREF
#define	LASSO_ECP_PREFIX
#define	LASSO_LIB_MAJOR_VERSION_N
#define	LASSO_LIB_MINOR_VERSION_N
#define	LASSO_SAML_MAJOR_VERSION_N
#define	LASSO_SAML_MINOR_VERSION_N
#define	LASSO_LIB_NAMEID_POLICY_TYPE_NONE
#define	LASSO_LIB_NAMEID_POLICY_TYPE_ONE_TIME
#define	LASSO_LIB_NAMEID_POLICY_TYPE_FEDERATED
#define	LASSO_LIB_NAMEID_POLICY_TYPE_ANY
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_INTERNET_PROTOCOL
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_INTERNET_PROTOCOL_PASSWOR
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_ONE_FACTOR_UNREGIST
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_TWO_FACTOR_UNREGIS
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_ONE_FACTOR_CONTRAC
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_TWO_FACTOR_CONTRA
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PASSWORD
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PASSWORD_PROTECTED_TRANSP
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PREVIOUS_SESSION
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SMARTCARD
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SMARTCARD_PKI
#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SOFTWARE_PKI

#define	LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_TIME_SYNC_TOKEN
#define	LASSO_LIB_AUTHN_CONTEXT_COMPARISON_EXACT
#define	LASSO_LIB_AUTHN_CONTEXT_COMPARISON_MINIMUM
#define	LASSO_LIB_AUTHN_CONTEXT_COMPARISON_MAXIMUM
#define	LASSO_LIB_AUTHN_CONTEXT_COMPARISON_BETTER
#define	LASSO_SAML_AUTHENTICATION_METHOD_PASSWORD
#define	LASSO_SAML_AUTHENTICATION_METHOD_KERBEROS
#define	LASSO_SAML_AUTHENTICATION_METHOD_SECURE_REMOTE_PASSWORD
#define	LASSO_SAML_AUTHENTICATION_METHOD_HARDWARE_TOKEN
#define	LASSO_SAML_AUTHENTICATION_METHOD_SMARTCARD_PKI
#define	LASSO_SAML_AUTHENTICATION_METHOD_SOFTWARE_PKI
#define	LASSO_SAML_AUTHENTICATION_METHOD_PGP
#define	LASSO_SAML_AUTHENTICATION_METHODS_PKI
#define	LASSO_SAML_AUTHENTICATION_METHOD_XKMS
#define	LASSO_SAML_AUTHENTICATION_METHOD_XMLD_SIG
#define	LASSO_SAML_AUTHENTICATION_METHOD_UNSPECIFIED
#define	LASSO_SAML_AUTHENTICATION_METHOD_LIBERTY
#define	LASSO_LIB_STATUS_CODE_FEDERATION_DOES_NOT_EXIST
#define	LASSO_LIB_STATUS_CODE_INVALID_ASSERTION_CONSUMER_SERVICE_INDE
#define	LASSO_LIB_STATUS_CODE_INVALID_SIGNATURE
#define	LASSO_LIB_STATUS_CODE_NO_AUTHN_CONTEXT
#define	LASSO_LIB_STATUS_CODE_NO_AVAILABLEIDP
#define	LASSO_LIB_STATUS_CODE_NO_PASSIVE
#define	LASSO_LIB_STATUS_CODE_NO_SUPPORTEDIDP
#define	LASSO_LIB_STATUS_CODE_PROXY_COUNT_EXCEEDED
#define	LASSO_LIB_STATUS_CODE_UNKNOWN_PRINCIPAL
#define	LASSO_LIB_STATUS_CODE_UNSIGNED_AUTHN_REQUEST
#define	LASSO_LIB_STATUS_CODE_UNSUPPORTED_PROFILE
#define	LASSO_SAML_STATUS_CODE_SUCCESS
#define	LASSO_SAML_STATUS_CODE_VERSION_MISMATCH
#define	LASSO_SAML_STATUS_CODE_REQUESTER
#define	LASSO_SAML_STATUS_CODE_RESPONDER
#define	LASSO_SAML_STATUS_CODE_REQUEST_VERSION_TOO_HIGH
#define	LASSO_SAML_STATUS_CODE_REQUEST_VERSION_TOO_LOW
#define	LASSO_SAML_STATUS_CODE_REQUEST_VERSION_DEPRECATED
#define	LASSO_SAML_STATUS_CODE_TOO_MANY_RESPONSES
#define	LASSO_SAML_STATUS_CODE_RESOURCE_NOT_RECOGNIZED
#define	LASSO_SAML_STATUS_CODE_REQUEST_DENIED
#define	LASSO_LIB_PROTOCOL_PROFILE_BRWS_ART
#define	LASSO_LIB_PROTOCOL_PROFILE_BRWS_POST
#define	LASSO_LIB_PROTOCOL_PROFILE_BRWS_LECP
#define	LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_IDP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_IDP_SOAP
#define	LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_SP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_SP_SOAP
#define	LASSO_LIB_PROTOCOL_PROFILE_NIM_SP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_RNI_IDP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_RNI_IDP_SOAP
#define	LASSO_LIB_PROTOCOL_PROFILE_RNI_SP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_RNI_SP_SOAP
#define	LASSO_LIB_PROTOCOL_PROFILE_SLO_SP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_SLO_SP_SOAP
#define	LASSO_LIB_PROTOCOL_PROFILE_SLO_IDP_HTTP
#define	LASSO_LIB_PROTOCOL_PROFILE_SLO_IDP_SOAP
#define	LASSO_LIB_NAME_IDENTIFIER_FORMAT_FEDERATED
#define	LASSO_LIB_NAME_IDENTIFIER_FORMAT_ONE_TIME

#define	LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENCRYPTED
#define	LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENTITYID
#define	LASSO_LIB_CONSENT_OBTAINED
#define	LASSO_LIB_CONSENT_OBTAINED_PRIOR
#define	LASSO_LIB_CONSENT_OBTAINED_CURRENT_IMPLICIT
#define	LASSO_LIB_CONSENT_OBTAINED_CURRENT_EXPLICIT
#define	LASSO_LIB_CONSENT_UNAVAILABLE
#define	LASSO_LIB_CONSENT_INAPPLICABLE
#define	LASSO_SAML_CONFIRMATION_METHOD_ARTIFACT
#define	LASSO_SAML_CONFIRMATION_METHOD_ARTIFACT01
#define	LASSO_SAML_CONFIRMATION_METHOD_BEARER
#define	LASSO_SAML_CONFIRMATION_METHOD_HOLDER_OF_KEY
#define	LASSO_SAML_CONFIRMATION_METHOD_SENDER_VOUCHES

Description

Functions

Types and Values

LASSO_LIB_HREF

```
#define LASSO_LIB_HREF   "urn:liberty:iff:2003-08"
```

Namespace for the elements specific to ID-FF 1.2 (not part of SAML 1.0)

LASSO_LIB_PREFIX

```
#define LASSO_LIB_PREFIX   "lib"
```

Preferred prefix for the ID-FF 1.2 namespace

LASSO_METADATA_HREF

```
#define LASSO_METADATA_HREF  "urn:liberty:metadata:2003-08"
```

Namespace for ID-FF 1.2 metadatas.

LASSO_METADATA_PREFIX

```
#define LASSO_METADATA_PREFIX  "md"
```

Preferred prefix for ID-FF 1.2 metadata namespace.

LASSO_SAML_ASSERTION_HREF

```
#define LASSO_SAML_ASSERTION_HREF~"urn:oasis:names:tc:SAML:1.0:assertion"
```

Namespace for SAML 1.0 assertion elements.

LASSO_SAML_ASSERTION_PREFIX

```
#define LASSO_SAML_ASSERTION_PREFIX~"saml"
```

Preferred prefix for assertion elements.

LASSO_SAML_PROTOCOL_HREF

```
#define LASSO_SAML_PROTOCOL_HREF~"urn:oasis:names:tc:SAML:1.0:protocol"
```

Namespace for SAML 1.0 protocol elements.

LASSO_SAML_PROTOCOL_PREFIX

```
#define LASSO_SAML_PROTOCOL_PREFIX~"samlp"
```

Preferred prefix for assertion elements.

LASSO_PAOS_HREF

```
#define LASSO_PAOS_HREF   "urn:liberty:paos:2003-08"
```

Namespace for FIXME

LASSO_PAOS_PREFIX

```
#define LASSO_PAOS_PREFIX "paos"
```

URN binding name for PAOS

LASSO_ECP_HREF

```
#define LASSO_ECP_HREF   "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
```

Namespace for FIXME

LASSO_ECP_PREFIX

```
#define LASSO_ECP_PREFIX "ecp"
```

SAML 2 URN profile name for Enhanced Client & Proxy (ECP)

LASSO_LIB_MAJOR_VERSION_N

```
#define LASSO_LIB_MAJOR_VERSION_N  1
```

Major version of the ID-FF protocol supported.

LASSO_LIB_MINOR_VERSION_N

```
#define LASSO_LIB_MINOR_VERSION_N  2
```

Minor version of the ID-FF protocol supported.

LASSO_SAML_MAJOR_VERSION_N

```
#define LASSO_SAML_MAJOR_VERSION_N   1
```

Major version number of the SAML specification used for ID-FF support in Lasso.

LASSO_SAML_MINOR_VERSION_N

```
#define LASSO_SAML_MINOR_VERSION_N   1
```

Minor version number of the SAML specification used for ID-FF support in Lasso.

LASSO_LIB_NAMEID_POLICY_TYPE_NONE

```
#define LASSO_LIB_NAMEID_POLICY_TYPE_NONE  "none"
```

None policy for use in LassoLibAuthnRequest. It means an existing federation must be used and an error should be produced if none existed beforehand.

LASSO_LIB_NAMEID_POLICY_TYPE_ONE_TIME

```
#define LASSO_LIB_NAMEID_POLICY_TYPE_ONE_TIME  "onetime"
```

Onetime policy for use in LassoLibAuthnRequest. It means a federation must not be created between identity and service provider. A temporary name identifier should be used instead.

LASSO_LIB_NAMEID_POLICY_TYPE_FEDERATED

```
#define LASSO_LIB_NAMEID_POLICY_TYPE_FEDERATED   "federated"
```

Federated policy for use in LassoLibAuthnRequest. It means a federation may be created between identity and service provider (if it didn't exist before).

LASSO_LIB_NAMEID_POLICY_TYPE_ANY

```
#define LASSO_LIB_NAMEID_POLICY_TYPE_ANY   "any"
```

Any policy for use in LassoLibAuthnRequest. It means a federation may be created if the principal agrees and it can fall back to *onetime* if he does not.

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_INTERNET_PROTOCOL

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_INTERNET_PROTOCOL
```

The Internet Protocol class is identified when a Principal is authenticated through the use of a provided IP address. Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_INTERNET_PROTOCOL_PASSWORD

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_INTERNET_PROTOCOL_PASSWORD
```

The Internet Protocol Password class is identified when a Principal is authenticated through the use of a provided IP address, in addition to username/password. Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_ONE_FACTOR_UNREGISTERED

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_ONE_FACTOR_UNREGISTERED
```

Reflects no mobile customer registration procedures and an authentication of the mobile device without requiring explicit end-user interaction. Again, this context authenticates only the device and never the user, it is useful when services other than the mobile operator want to add a secure device authentication to their authentication process.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_TWO_FACTOR_UNREGISTERED

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_TWO_FACTOR_UNREGISTERED
```

Reflects no mobile customer registration procedures and a two-factor based authentication, such as secure device and user PIN. This context class is useful when a service other than the mobile operator wants to link their customer ID to a mobile supplied two-factor authentication service by capturing mobile phone data at enrollment.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_ONE_FACTOR_CONTRACT

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_ONE_FACTOR_CONTRACT
```

Reflects mobile contract customer registration procedures and a single factor authentication. For example, a digital signing device with tamper resistant memory for key storage, such as the mobile MSISDN, but no required PIN or biometric for real-time user authentication.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_TWO_FACTOR_CONTRACT

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_MOBILE_TWO_FACTOR_CONTRACT
```

Reflects mobile contract customer registration procedures and a two-factor based authentication. For example, a digital signing device with tamper resistant memory for key storage, such as a GSM SIM, that requires explicit proof of user identity and intent, such as a PIN or biometric.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PASSWORD

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PASSWORD
```

The Password class is identified when a Principal authenticates to an identity provider through the presentation of a password over an unprotected HTTP session.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PASSWORD_PROTECTED_TRANSPORT

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PASSWORD_PROTECTED_TRANSPORT
```

The PasswordProtectedTransport class is identified when a Principal authenticates to an identity provider through the presentation of a password over a protected session.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PREVIOUS_SESSION

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_PREVIOUS_SESSION
```

The PreviousSession class is identified when a Principal had authenticated to an identity provider at some point in the past using any authentication context supported by that identity provider. Consequently, a subsequent authentication event that the identity provider will assert to the service provider may be significantly separated in time from the Principals current resource access request. The context for the previously authenticated session is explicitly not included in this context class because the user has not authenticated during this session, and so the mechanism that the user employed to authenticate in a previous session should not be used as part of a decision on whether to now allow access to a resource.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SMARTCARD

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SMARTCARD
```

The Smartcard class is identified when a Principal authenticates to an identity provider using a smartcard.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SMARTCARD_PKI

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SMARTCARD_PKI
```

The SmartcardPKI class is identified when a Principal authenticates to an identity provider through a two-factor authentication mechanism using a smartcard with enclosed private key and a PIN.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SOFTWARE_PKI

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_SOFTWARE_PKI
```

The Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to authenticate to the identity provider.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_TIME_SYNC_TOKEN

```
#define             LASSO_LIB_AUTHN_CONTEXT_CLASS_REF_TIME_SYNC_TOKEN
```

The TimeSyncToken class is identified when a Principal authenticates through a time synchronization token.

Source: Liberty ID-FF Authentication Context Specification v1.3

LASSO_LIB_AUTHN_CONTEXT_COMPARISON_EXACT

```
#define LASSO_LIB_AUTHN_CONTEXT_COMPARISON_EXACT   "exact"
```

Ask for the exact authentication context.

LASSO_LIB_AUTHN_CONTEXT_COMPARISON_MINIMUM

```
#define LASSO_LIB_AUTHN_CONTEXT_COMPARISON_MINIMUM   "minimum"
```

Ask for at least this authentication context.

LASSO_LIB_AUTHN_CONTEXT_COMPARISON_MAXIMUM

```
#define LASSO_LIB_AUTHN_CONTEXT_COMPARISON_MAXIMUM   "maximum"
```

Ask for at most this authentication context.

LASSO_LIB_AUTHN_CONTEXT_COMPARISON_BETTER

```
#define LASSO_LIB_AUTHN_CONTEXT_COMPARISON_BETTER  "better"
```

Ask for a better authentication context than that.

LASSO_SAML_AUTHENTICATION_METHOD_PASSWORD

```
#define LASSO_SAML_AUTHENTICATION_METHOD_PASSWORD  "urn:oasis:names:tc:SAML:1.0:am:password ↩
    "
```

The authentication was performed by means of a password.

LASSO_SAML_AUTHENTICATION_METHOD_KERBEROS

```
#define LASSO_SAML_AUTHENTICATION_METHOD_KERBEROS  "urn:ietf:rfc:1510"
```

The authentication was performed by means of the Kerberos protocol [RFC 1510], an instantiation of the Needham-Schroeder symmetric key authentication mechanism [Needham78].

LASSO_SAML_AUTHENTICATION_METHOD_SECURE_REMOTE_PASSWORD

```
#define LASSO_SAML_AUTHENTICATION_METHOD_SECURE_REMOTE_PASSWORD  "urn:ietf:rfc:2945"
```

The authentication was performed by means of Secure Remote Password protocol as specified in [RFC 2945].

LASSO_SAML_AUTHENTICATION_METHOD_HARDWARE_TOKEN

```
#define             LASSO_SAML_AUTHENTICATION_METHOD_HARDWARE_TOKEN
```

The authentication was performed using some (unspecified) hardware token.

LASSO_SAML_AUTHENTICATION_METHOD_SMARTCARD_PKI

```
#define LASSO_SAML_AUTHENTICATION_METHOD_SMARTCARD_PKI  "urn:ietf:rfc:2246"
```

The authentication was performed using either the SSL or TLS protocol with certificate-based client authentication. TLS is described in [RFC 2246].

LASSO_SAML_AUTHENTICATION_METHOD_SOFTWARE_PKI

```
#define LASSO_SAML_AUTHENTICATION_METHOD_SOFTWARE_PKI   "urn:oasis:names:tc:SAML:1.0:am: ←
    X509-PKI"
```

The authentication was performed by some (unspecified) mechanism on a key authenticated by means of an X.509 PKI X.500. It may have been one of the mechanisms for which a more specific identifier has been defined below.

LASSO_SAML_AUTHENTICATION_METHOD_PGP

```
#define LASSO_SAML_AUTHENTICATION_METHOD_PGP            "urn:oasis:names:tc:SAML:1.0:am:PGP ←
    "
```

The authentication was performed by some (unspecified) mechanism on a key authenticated by means of a PGP web of trust [PGP]. It may have been one of the mechanisms for which a more specific identifier has been defined below.

LASSO_SAML_AUTHENTICATION_METHODS_PKI

```
#define LASSO_SAML_AUTHENTICATION_METHODS_PKI           "urn:oasis:names:tc:SAML:1.0:am: ←
    SPKI"
```

The authentication was performed by some (unspecified) mechanism on a key authenticated by means of a PGP web of trust [PGP]. It may have been one of the mechanisms for which a more specific identifier has been defined below.

LASSO_SAML_AUTHENTICATION_METHOD_XKMS

```
#define LASSO_SAML_AUTHENTICATION_METHOD_XKMS           "urn:oasis:names:tc:SAML:1.0:am: ←
    XKMS"
```

The authentication was performed by some (unspecified) mechanism on a key authenticated by means of a PGP web of trust [PGP]. It may have been one of the mechanisms for which a more specific identifier has been defined below.

LASSO_SAML_AUTHENTICATION_METHOD_XMLD_SIG

```
#define LASSO_SAML_AUTHENTICATION_METHOD_XMLD_SIG       "urn:ietf:rfc:3075"
```

The authentication was performed by means of an XML digital signature [RFC 3075].

LASSO_SAML_AUTHENTICATION_METHOD_UNSPECIFIED

```
#define             LASSO_SAML_AUTHENTICATION_METHOD_UNSPECIFIED
```

The authentication was performed by an unspecified means.

LASSO_SAML_AUTHENTICATION_METHOD_LIBERTY

```
#define LASSO_SAML_AUTHENTICATION_METHOD_LIBERTY        "urn:liberty:ac:2003-08"
```

The authentication was performed by a liberty alliance protocol.

LASSO_LIB_STATUS_CODE_FEDERATION_DOES_NOT_EXIST

```
#define LASSO_LIB_STATUS_CODE_FEDERATION_DOES_NOT_EXIST    "lib:FederationDoesNotExist"
```

Second level status code.

Used by an identity provider to indicate that the Principal has not federated his or her identity with the service provider, and the service provider indicated a requirement for federation.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_INVALID_ASSERTION_CONSUMER_SERVICE_INDEX

```
#define             LASSO_LIB_STATUS_CODE_INVALID_ASSERTION_CONSUMER_SERVICE_INDEX
```

Second level status code.

If the <AssertionConsumerServiceID> element is provided, then the identity provider *MUST* search for the value among the id attributes in the <AssertionConsumerServiceURL> elements in the provider's meta- data to determine the URL to use. If no match can be found, then the provider *MUST* return an error with a second-level <samlp:StatusCode> of lib:InvalidAssertionConsumerServiceI
to the default URL (the <AssertionConsumerServiceURL> with an isDefault attribute of "true").

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_INVALID_SIGNATURE

```
#define LASSO_LIB_STATUS_CODE_INVALID_SIGNATURE            "lib:InvalidSignature"
```

Second level status code.

Indicate a failure in the processing of the signature of the request. This code is not part of the ID-FF 1.2 specification.

LASSO_LIB_STATUS_CODE_NO_AUTHN_CONTEXT

```
#define LASSO_LIB_STATUS_CODE_NO_AUTHN_CONTEXT             "lib:NoAuthnContext"
```

Used by an identity provider to indicate that the specified authentication context information in the request prohibits authentication from taking place.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_NO_AVAILABLEIDP

```
#define LASSO_LIB_STATUS_CODE_NO_AVAILABLEIDP              "lib:NoAvailableIDP"
```

Used by an intermediary to indicate that none of the supported identity provider URLs from the <IDPList> can be resolved or that none of the supported identity providers are available.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_NO_PASSIVE

```
#define LASSO_LIB_STATUS_CODE_NO_PASSIVE                   "lib:NoPassive"
```

Used by an identity provider or an intermediary to indicate that authentication of the Principal requires interaction and cannot be performed passively.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_NO_SUPPORTEDIDP

```
#define LASSO_LIB_STATUS_CODE_NO_SUPPORTEDIDP              "lib:NoSupportedIDP"
```

Used by an intermediary to indicate that none of the identity providers are supported by the intermediary.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_PROXY_COUNT_EXCEEDED

```
#define LASSO_LIB_STATUS_CODE_PROXY_COUNT_EXCEEDED         "lib:ProxyCountExceeded"
```

Used by an identity provider to indicate that it cannot authenticate the principal itself, and was not permitted to relay the request further.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_UNKNOWN_PRINCIPAL

```
#define LASSO_LIB_STATUS_CODE_UNKNOWN_PRINCIPAL            "lib:UnknownPrincipal"
```

Used by an identity provider to indicate that the Principal is not known to it.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_UNSIGNED_AUTHN_REQUEST

```
#define LASSO_LIB_STATUS_CODE_UNSIGNED_AUTHN_REQUEST       "lib:UnsignedAuthnRequest"
```

If the requesting provider's <AuthnRequestsSigned> metadata element is "true", then any request messages it generates *MUST* be signed. If an unsigned request is received, then the provider *MUST* return an error with a second- level <samlp:StatusCode> of lib:UnsignedAuthnRequest.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_LIB_STATUS_CODE_UNSUPPORTED_PROFILE

```
#define LASSO_LIB_STATUS_CODE_UNSUPPORTED_PROFILE          "lib:UnsupportedProfile"
```

If an error occurs during this further processing of the logout (for example, relying service providers may not all implement the Single Logout profile used by the requesting service provider), then the identity provider *MUST* respond to the original requester with a <LogoutResponse> message, indicating the status of the logout request. The value "lib:UnsupportedProfile" is provided for a second-level <samlp:StatusCode>, indicating that a service provider should retry the <LogoutRequest> using a different profile.

Source: Liberty ID-FF Protocols and Schema Specification 1.2

LASSO_SAML_STATUS_CODE_SUCCESS

```
#define LASSO_SAML_STATUS_CODE_SUCCESS              "samlp:Success"
```

A protocol request succeeded.

LASSO_SAML_STATUS_CODE_VERSION_MISMATCH

```
#define LASSO_SAML_STATUS_CODE_VERSION_MISMATCH          "samlp:VersionMismatch"
```

Request failed, because the version is not supported by the provider. Look at second level status for more details.

LASSO_SAML_STATUS_CODE_REQUESTER

```
#define LASSO_SAML_STATUS_CODE_REQUESTER                "samlp:Requester"
```

Request failed because of the requester. Look at second level status for more details.

LASSO_SAML_STATUS_CODE_RESPONDER

```
#define LASSO_SAML_STATUS_CODE_RESPONDER                "samlp:Responder"
```

Request failed because of the responder. Look at second level status for more details.

LASSO_SAML_STATUS_CODE_REQUEST_VERSION_TOO_HIGH

```
#define LASSO_SAML_STATUS_CODE_REQUEST_VERSION_TOO_HIGH    "samlp:RequestVersionTooHigh"
```

Request failed because the version of protocol used is too high. Used with LASSO_SAML_STATUS_CODE_VERSION_MISMATCH.

LASSO_SAML_STATUS_CODE_REQUEST_VERSION_TOO_LOW

```
#define LASSO_SAML_STATUS_CODE_REQUEST_VERSION_TOO_LOW     "samlp:RequestVersionTooLow"
```

Request failed because the version of protocol used is too low. Used with LASSO_SAML_STATUS_CODE_VERSION_MISMATCH.

LASSO_SAML_STATUS_CODE_REQUEST_VERSION_DEPRECATED

```
#define LASSO_SAML_STATUS_CODE_REQUEST_VERSION_DEPRECATED "samlp:RequestVersionDeprecated"
```

Request failed because the version of protocol used is deprecated. Used with LASSO_SAML_STATUS_CODE_VERSION_MISMATC

LASSO_SAML_STATUS_CODE_TOO_MANY_RESPONSES

```
#define LASSO_SAML_STATUS_CODE_TOO_MANY_RESPONSES         "samlp:TooManyResponses"
```

Request failed because too many data should be returned. Used with LASSO_SAML_STATUS_CODE_RESPONDER.

LASSO_SAML_STATUS_CODE_RESOURCE_NOT_RECOGNIZED

```
#define LASSO_SAML_STATUS_CODE_RESOURCE_NOT_RECOGNIZED    "samlp:ResourceNotRecognized"
```

Request failed because the responder does not wish to support resource-specific attribute queries, or the resource value provided is invalid or unrecognized. Use with LASSO_SAML_STATUS_CODE_RESPONDER.

LASSO_SAML_STATUS_CODE_REQUEST_DENIED

```
#define LASSO_SAML_STATUS_CODE_REQUEST_DENIED            "samlp:RequestDenied"
```

The SAML responder or SAML authority is able to process the request but has chosen not to respond. This status code MAY be used when there is concern about the security context of the request message or the sequence of request messages received from a particular requester.

Source: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1

LASSO_LIB_PROTOCOL_PROFILE_BRWS_ART

```
#define             LASSO_LIB_PROTOCOL_PROFILE_BRWS_ART
```

Identifies the Single Sign-On "Artifact" profile; where an artifact is passed from identity provider to service provider and back to get the LassoLibAssertion.

LASSO_LIB_PROTOCOL_PROFILE_BRWS_POST

```
#define             LASSO_LIB_PROTOCOL_PROFILE_BRWS_POST
```

Identifies the Single Sign-On "POST" profile; where the LassoLibAssertion is sent directly from the identity provider to the service provider in an HTML form submission message.

LASSO_LIB_PROTOCOL_PROFILE_BRWS_LECP

```
#define             LASSO_LIB_PROTOCOL_PROFILE_BRWS_LECP
```

Identifies the Single Sign-On "LECP" profile; where the LassoLibAssertion is sent directly from the identity provider to the service provider in a PAOS response. See LassoLecp.

LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_IDP_HTTP

```
#define             LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_IDP_HTTP
```

Identifies the Federation Termination "Redirect" profile; where the request for federation termination is sent from the identity provider to the service provider in a redirected GET request.

LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_IDP_SOAP

```
#define             LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_IDP_SOAP
```

LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_SP_HTTP

```
#define              LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_SP_HTTP
```

LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_SP_SOAP

```
#define              LASSO_LIB_PROTOCOL_PROFILE_FED_TERM_SP_SOAP
```

LASSO_LIB_PROTOCOL_PROFILE_NIM_SP_HTTP

```
#define              LASSO_LIB_PROTOCOL_PROFILE_NIM_SP_HTTP
```

LASSO_LIB_PROTOCOL_PROFILE_RNI_IDP_HTTP

```
#define LASSO_LIB_PROTOCOL_PROFILE_RNI_IDP_HTTP "http://projectliberty.org/profiles/rni-idp ↩
    -http"
```

LASSO_LIB_PROTOCOL_PROFILE_RNI_IDP_SOAP

```
#define LASSO_LIB_PROTOCOL_PROFILE_RNI_IDP_SOAP "http://projectliberty.org/profiles/rni-idp ↩
    -soap"
```

LASSO_LIB_PROTOCOL_PROFILE_RNI_SP_HTTP

```
#define LASSO_LIB_PROTOCOL_PROFILE_RNI_SP_HTTP  "http://projectliberty.org/profiles/rni-sp- ↩
    http"
```

LASSO_LIB_PROTOCOL_PROFILE_RNI_SP_SOAP

```
#define LASSO_LIB_PROTOCOL_PROFILE_RNI_SP_SOAP  "http://projectliberty.org/profiles/rni-sp- ↩
    soap"
```

LASSO_LIB_PROTOCOL_PROFILE_SLO_SP_HTTP

```
#define LASSO_LIB_PROTOCOL_PROFILE_SLO_SP_HTTP  "http://projectliberty.org/profiles/slo-sp- ↩
    http"
```

LASSO_LIB_PROTOCOL_PROFILE_SLO_SP_SOAP

```
#define LASSO_LIB_PROTOCOL_PROFILE_SLO_SP_SOAP  "http://projectliberty.org/profiles/slo-sp- ↩
    soap"
```

LASSO_LIB_PROTOCOL_PROFILE_SLO_IDP_HTTP

```
#define LASSO_LIB_PROTOCOL_PROFILE_SLO_IDP_HTTP "http://projectliberty.org/profiles/slo-idp ↩
    -http"
```

LASSO_LIB_PROTOCOL_PROFILE_SLO_IDP_SOAP

```
#define LASSO_LIB_PROTOCOL_PROFILE_SLO_IDP_SOAP "http://projectliberty.org/profiles/slo-idp ↩
    -soap"
```

LASSO_LIB_NAME_IDENTIFIER_FORMAT_FEDERATED

```
#define LASSO_LIB_NAME_IDENTIFIER_FORMAT_FEDERATED "urn:liberty:iff:nameid:federated"
```

Federated name identifier constant, used in LassoSamlNameIdentifier. It implies the name identifier belongs to a federation established between SP and IdP.

LASSO_LIB_NAME_IDENTIFIER_FORMAT_ONE_TIME

```
#define LASSO_LIB_NAME_IDENTIFIER_FORMAT_ONE_TIME  "urn:liberty:iff:nameid:one-time"
```

"One-time" name identifier constant, used in LassoSamlNameIdentifier.

LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENCRYPTED

```
#define LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENCRYPTED "urn:liberty:iff:nameid:encrypted"
```

"Encrypted" name identifier constant, used in LassoSamlNameIdentifier.

LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENTITYID

```
#define LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENTITYID  "urn:liberty:iff:nameid:entityID"
```

LASSO_LIB_CONSENT_OBTAINED

```
#define LASSO_LIB_CONSENT_OBTAINED                  "urn:liberty:consent:obtained"
```

LASSO_LIB_CONSENT_OBTAINED_PRIOR

```
#define LASSO_LIB_CONSENT_OBTAINED_PRIOR            "urn:liberty:consent:obtained:prior"
```

LASSO_LIB_CONSENT_OBTAINED_CURRENT_IMPLICIT

```
#define LASSO_LIB_CONSENT_OBTAINED_CURRENT_IMPLICIT "urn:liberty:consent:obtained:current: ↩
    implicit"
```

LASSO_LIB_CONSENT_OBTAINED_CURRENT_EXPLICIT

```
#define LASSO_LIB_CONSENT_OBTAINED_CURRENT_EXPLICIT "urn:liberty:consent:obtained:current: ↩
    explicit"
```

LASSO_LIB_CONSENT_UNAVAILABLE

```
#define LASSO_LIB_CONSENT_UNAVAILABLE              "urn:liberty:consent:unavailable"
```

LASSO_LIB_CONSENT_INAPPLICABLE

```
#define LASSO_LIB_CONSENT_INAPPLICABLE             "urn:liberty:consent:inapplicable"
```

LASSO_SAML_CONFIRMATION_METHOD_ARTIFACT

```
#define LASSO_SAML_CONFIRMATION_METHOD_ARTIFACT "urn:oasis:names:tc:SAML:1.0:cm:artifact"
```

Confirmation method when the browser-artifact binding is used.

LASSO_SAML_CONFIRMATION_METHOD_ARTIFACT01

```
#define LASSO_SAML_CONFIRMATION_METHOD_ARTIFACT01 "urn:oasis:names:tc:SAML:1.0:cm:artifact ↩
    -01"
```

Deprecated confirmation method when the browser-artifact binding is used.

LASSO_SAML_CONFIRMATION_METHOD_BEARER

```
#define LASSO_SAML_CONFIRMATION_METHOD_BEARER "urn:oasis:names:tc:SAML:1.0:cm:bearer"
```

Confirmation method when subject of the assertion is the one holding it.

LASSO_SAML_CONFIRMATION_METHOD_HOLDER_OF_KEY

```
#define             LASSO_SAML_CONFIRMATION_METHOD_HOLDER_OF_KEY
```

A ds:KeyInfo must be present in the SubjecConfirmation element. It *MUST* be used to confirm assertion subject identity.

LASSO_SAML_CONFIRMATION_METHOD_SENDER_VOUCHES

```
#define             LASSO_SAML_CONFIRMATION_METHOD_SENDER_VOUCHES
```

Indicates that no other information is available about the context of use of the assertion. The relying party *SHOULD* utilize other means to determine if it should process the assertion further.

5.2 LassoLibAssertion

LassoLibAssertion — <lib:Assertion>

Functions

LassoLibAssertion *	lasso_lib_assertion_new ()
LassoLibAssertion *	lasso_lib_assertion_new_full ()

Types and Values

struct	LassoLibAssertion

Description

```
<xs:element name="Assertion" type="AssertionType" substitutionGroup="saml:Assertion" />
<xs:complexType name="AssertionType">
  <xs:complexContent>
    <xs:extension base="saml:AssertionType">
      <xs:attribute name="InResponseTo" type="xs:NCName" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 5.1: Schema fragment for lib:Assertion

Functions

lasso_lib_assertion_new ()

```
LassoLibAssertion~*
lasso_lib_assertion_new (void);
```

Creates a new LassoLibAssertion object.

Returns

a newly created LassoLibAssertion object

lasso_lib_assertion_new_full ()

```
LassoLibAssertion~*
lasso_lib_assertion_new_full (const char *issuer,
                              const char *requestID,
                              const char *audience,
                              const char *notBefore,
                              const char *notOnOrAfter);
```

Creates a new LassoLibAssertion object and initializes its Issuer, InResponseTo, AudienceRestrictionCondition, notBefore and notOnOrAfter fields or attributes.

Parameters

issuer	the issuer entityID string	
requestID	the identifier of the request which initiated the creation of this assertion.	*[allow-none]*
audience	the entityID of the receiver of this assertion.	*[allow-none]*
notBefore	a timestamp formatted as iso-8601	
notOnOrAfter	a timestamp formatted as iso-8601	

Returns

a newly created LassoLibAssertion object

Types and Values

struct LassoLibAssertion

```
struct LassoLibAssertion {
  LassoSamlAssertion parent;

  char *InResponseTo;
};
```

5.3 LassoLibAuthenticationStatement

LassoLibAuthenticationStatement — <lib:AuthenticationStatement>

Functions

LassoLibAuthenticationStatement *	lasso_lib_authentication_statement_new ()
LassoLibAuthenticationStatement *	lasso_lib_authentication_statement_new_full ()

Types and Values

struct	LassoLibAuthenticationStatement

Description

```
<xs:element name="AuthenticationStatement" type="AuthenticationStatementType"
  substitutionGroup="saml:Statement"/>
<xs:complexType name="AuthenticationStatementType">
  <xs:complexContent>
    <xs:extension base="saml:AuthenticationStatementType">
      <xs:sequence>
        <xs:element ref="AuthnContext" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="ReauthenticateOnOrAfter" type="xs:dateTime" use="optional"/>
      <xs:attribute name="SessionIndex" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 5.2: Schema fragment for lib:AuthenticationStatement

Functions

lasso_lib_authentication_statement_new ()

```
LassoLibAuthenticationStatement~*
lasso_lib_authentication_statement_new
                                (void);
```

Creates a new LassoLibAuthenticationStatement object.

Returns

a newly created LassoLibAuthenticationStatement object

lasso_lib_authentication_statement_new_full ()

```
LassoLibAuthenticationStatement~*
lasso_lib_authentication_statement_new_full
                                (const char *authenticationMethod,
                                 const char *authenticationInstant,
                                 const char *reauthenticateOnOrAfter,
                                 LassoSamlNameIdentifier *sp_identifier,
                                 LassoSamlNameIdentifier *idp_identifier);
```

Creates a new LassoLibAuthenticationStatement object and initializes its subject, its AuthenticationMethod, its AuthenticationInstant,

Parameters

authenticationMethod	an URI identifier for the authentication method	
authenticationInstant	an ISO-8601 formatted timestamp for the authentication instant.	*[allow-none]*

reauthenticateOnOrAfter	an ISO-8601 formatted timestamp to set a limit on the value of this authentication.	*[allow-none]*
sp_identifier	(allow-none) a LassoSamlNameIdentifier object, the SP qualifier for the subject of this statement	
idp_identifier	a LassoSamlNameIdentifier object, the IdP qualifier for the subject of this statemtn	

Returns

a newly created LassoLibAuthenticationStatement object

Types and Values

struct LassoLibAuthenticationStatement

```
struct LassoLibAuthenticationStatement {
  LassoSamlAuthenticationStatement parent;

  /* <xs:element ref="AuthnContext" minOccurs="0"/> */
  LassoLibAuthnContext *AuthnContext;
  /* <xs:attribute name="ReauthenticateOnOrAfter" type="xs:dateTime" use="optional"/> */
  char *ReauthenticateOnOrAfter;
  /* <xs:attribute name="SessionIndex" type="xs:string" use="required"/> */
  char *SessionIndex;
};
```

5.4 LassoLibAuthnContext

LassoLibAuthnContext — <lib:AuthnContext>

Functions

LassoNode *	lasso_lib_authn_context_new ()

Types and Values

struct	LassoLibAuthnContext

Description

```
<xs:element name="AuthnContext">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AuthnContextClassRef" type="xs:anyURI" minOccurs="0"/>
      <xs:choice>
        <xs:element ref="ac:AuthenticationContextStatement"/>
        <xs:element name="AuthnContextStatementRef" type="xs:anyURI"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 5.3: Schema fragment for lib:AuthnContext

Functions

lasso_lib_authn_context_new ()

```
LassoNode~*
lasso_lib_authn_context_new (void);
```

Creates a new LassoLibAuthnContext object.

Returns

a newly created LassoLibAuthnContext object

Types and Values

struct LassoLibAuthnContext

```
struct LassoLibAuthnContext {
  LassoNode parent;

  /* <xs:element name="AuthnContextClassRef" type="xs:anyURI" minOccurs="0"/> */
  char *AuthnContextClassRef;
  /* <xs:element name="AuthnContextStatementRef" type="xs:anyURI"/> */
  char *AuthnContextStatementRef;
  /* <xs:element ref="ac:AuthenticationContextStatement"/> */
  void *AuthenticationContextStatement; /* TODO: support that element */
};
```

5.5 LassoLibAuthnRequestEnvelope

LassoLibAuthnRequestEnvelope — <lib:AuthnRequestEnvelope>

Functions

LassoLibAuthnRequestEnvelope *	lasso_lib_authn_request_envelope_new ()
LassoLibAuthnRequestEnvelope *	lasso_lib_authn_request_envelope_new_full ()

Types and Values

struct	LassoLibAuthnRequestEnvelope

Description

```
<xs:element name="AuthnRequestEnvelope" type="AuthnRequestEnvelopeType"/>
<xs:complexType name="AuthnRequestEnvelopeType">
  <xs:complexContent>
    <xs:extension base="RequestEnvelopeType">
      <xs:sequence>
        <xs:element ref="AuthnRequest"/>
        <xs:element ref="ProviderID"/>
        <xs:element name="ProviderName" type="xs:string" minOccurs="0"/>
        <xs:element name="AssertionConsumerServiceURL" type="xs:anyURI"/>
        <xs:element ref="IDPList" minOccurs="0"/>
        <xs:element name="IsPassive" type="xs:boolean" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="RequestEnvelopeType">
  <xs:sequence>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="IDPList" type="IDPListType"/>
<xs:complexType name="IDPListType">
  <xs:sequence>
    <xs:element ref="IDPEntries"/>
    <xs:element ref="GetComplete" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ResponseEnvelopeType">
  <xs:sequence>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Figure 5.4: Schema fragment for lib:AuthnRequestEnvelope

Functions

lasso_lib_authn_request_envelope_new ()

```
LassoLibAuthnRequestEnvelope~*
lasso_lib_authn_request_envelope_new (void);
```

Creates a new LassoLibAuthnRequestEnvelope object.

Returns

a newly created LassoLibAuthnRequestEnvelope object

lasso_lib_authn_request_envelope_new_full ()

```
LassoLibAuthnRequestEnvelope~*
lasso_lib_authn_request_envelope_new_full
                                (LassoLibAuthnRequest *authnRequest,
                                 char *providerID,
                                 char *assertionConsumerServiceURL);
```

Creates a new LassoLibAuthnRequestEnvelope object and initializes it with the parameters.

Parameters

authnRequest	the LassoLibAuthnRequest to envelop	
providerID	service provider ID	
assertionConsumerServiceURL	assertion consumer service URL on the service provider	

Returns

a newly created LassoLibAuthnRequestEnvelope object

Types and Values

struct LassoLibAuthnRequestEnvelope

```
struct LassoLibAuthnRequestEnvelope {
  LassoNode parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="AuthnRequest"/> */
  LassoLibAuthnRequest *AuthnRequest;
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element name="ProviderName" type="xs:string" minOccurs="0"/> */
  char *ProviderName;
  /* <xs:element name="AssertionConsumerServiceURL" type="xs:anyURI"/> */
  char *AssertionConsumerServiceURL;
  /* <xs:element ref="IDPList" minOccurs="0"/> */
  LassoLibIDPList *IDPList;
  /* <xs:element name="IsPassive" type="xs:boolean" minOccurs="0"/> */
  gboolean IsPassive;
};
```

5.6 LassoLibAuthnRequest

LassoLibAuthnRequest — <lib:AuthnRequest>

Functions

LassoLibAuthnRequest *	lasso_lib_authn_request_new ()

Types and Values

struct	LassoLibAuthnRequest

Description

Authentication requests are sent from a service provider to an identity provider.

```
<xs:element name="AuthnRequest" type="AuthnRequestType" />
<xs:complexType name="AuthnRequestType">
  <xs:complexContent>
    <xs:extension base="samlp:RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ProviderID"/>
        <xs:element ref="AffiliationID" minOccurs="0"/>
        <xs:element ref="NameIDPolicy" minOccurs="0"/>
        <xs:element name="ForceAuthn" type="xs:boolean" minOccurs="0"/>
        <xs:element name="IsPassive" type="xs:boolean "minOccurs="0"/>
        <xs:element ref="ProtocolProfile" minOccurs="0"/>
        <xs:element name="AssertionConsumerServiceID" type="xs:string" minOccurs="0"/>
        <xs:element ref="RequestAuthnContext" minOccurs="0"/>
        <xs:element ref="RelayState" minOccurs="0"/>
        <xs:element ref="Scoping" minOccurs="0 "/>
      </xs:sequence>
      <xs:attribute ref="consent" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ProviderID" type="md:entityIDType"/>
<xs:element name="AffiliationID" type="md:entityIDType"/>

<xs:element name="NameIDPolicy" type="NameIDPolicyType"/>
<xs:simpleType name="NameIDPolicyType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none"/>
    <xs:enumeration value="onetime"/>
    <xs:enumeration value="federated"/>
    <xs:enumeration value="any"/ >
  </xs:restriction>
</xs:simpleType>

<xs:element name="ProtocolProfile" type="xs:anyURI"/>
<xs:element name="RelayState" type="xs:string"/>
```

Figure 5.5: Schema fragment for lib:AuthnRequest

Functions

lasso_lib_authn_request_new ()

```
LassoLibAuthnRequest~*
lasso_lib_authn_request_new (void);
```

Creates a new LassoLibAuthnRequest object.

Returns

a newly created LassoLibAuthnRequest object

Types and Values

struct LassoLibAuthnRequest

```
struct LassoLibAuthnRequest {
  LassoSamlpRequestAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="AffiliationID" minOccurs="0"/> */
  char *AffiliationID;
  /* <xs:element ref="NameIDPolicy" minOccurs="0"/> */
  char *NameIDPolicy;
  /* <xs:element name="ForceAuthn" type="xs:boolean" minOccurs="0"/> */
  gboolean ForceAuthn;
  /* <xs:element name="IsPassive" type="xs:boolean "minOccurs="0"/> */
  gboolean IsPassive;
  /* <xs:element ref="ProtocolProfile" minOccurs="0"/> */
  char *ProtocolProfile;
  /* <xs:element name="AssertionConsumerServiceID" type="xs:string" minOccurs="0"/> */
  char *AssertionConsumerServiceID;
  /* <xs:element ref="RequestAuthnContext" minOccurs="0"/> */
  LassoLibRequestAuthnContext *RequestAuthnContext;
  /* <xs:element ref="RelayState" minOccurs="0"/> */
  char *RelayState;
  /* <xs:element ref="Scoping" minOccurs="0 "/> */
  LassoLibScoping *Scoping;
  /* <xs:attribute ref="consent" use="optional"/> */
  char *consent;
};
```

`ProviderID` isthe service provider identifier, this field will often be filled with lasso_login_init_authn_request()

`nameIDPolicy` tells the identity provider about the policy to use for federation; it must be one of LASSO_LIB_NAMEID_POLICY_TY LASSO_LIB_NAMEID_POLICY_TYPE_ONE_TIME, LASSO_LIB_NAMEID_POLICY_TYPE_FEDERATED or LASSO_LIB_NA

`IsPassive` ; if TRUE (default) it tells the identity provider not to interact with the user.

`ForceAuthn` ; only used if *`IsPassive`* is FALSE, it tells the identity provider to force authentication of the user even when already authenticated.

`ProtocolProfile` is the Single Sign-On and Federation profile to adopt; either LASSO_LIB_PROTOCOL_PROFILE_BRWS_ART (which is the default value) or LASSO_LIB_PROTOCOL_PROFILE_BRWS_POST.

See Also

LassoLogin

5.7 LassoLibAuthnResponseEnvelope

LassoLibAuthnResponseEnvelope — <lib:AuthnResponseEnvelope>

Functions

LassoLibAuthnResponseEnvelope *	lasso_lib_authn_response_envelope_new ()

Types and Values

struct	LassoLibAuthnResponseEnvelope

Description

Functions

lasso_lib_authn_response_envelope_new ()

```
LassoLibAuthnResponseEnvelope~*
lasso_lib_authn_response_envelope_new (LassoLibAuthnResponse *response,
                                       char *assertionConsumerServiceURL);
```

Creates a new LassoLibAuthnResponseEnvelope object and initializes it with the parameters.

Parameters

response	the LassoLibAuthnResponse to envelop
assertionConsumerServiceURL	assertion consumer service URL on the service provider

Returns

a newly created LassoLibAuthnResponseEnvelope object

Types and Values

struct LassoLibAuthnResponseEnvelope

```
struct LassoLibAuthnResponseEnvelope {
  LassoNode parent;

  GList *Extension; /* of xmlNode* */
  LassoLibAuthnResponse *AuthnResponse;
  char *AssertionConsumerServiceURL;
};
```

5.8 LassoLibAuthnResponse

LassoLibAuthnResponse — <lib:AuthnResponse>

Functions

LassoNode *	lasso_lib_authn_response_new ()

Types and Values

struct	LassoLibAuthnResponse

Description

```
<xs:element name="AuthnResponse" type="AuthnResponseType"/>
<xs:complexType name="AuthnResponseType">
  <xs:complexContent>
    <xs:extension base="samlp:ResponseType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="ProviderID"/>
  <xs:element ref="RelayState" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute ref="consent" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ProviderID" type="md:entityIDType"/>
<xs:element name="RelayState" type="xs:string"/>
```

Figure 5.6: Schema fragment for lib:AuthnResponse

Functions

lasso_lib_authn_response_new ()

```
LassoNode~*
lasso_lib_authn_response_new (char *providerID,
                              LassoLibAuthnRequest *request);
```

Creates a new LassoLibAuthnResponse object.

Parameters

providerID	the identity provider ID	
request	the LassoLibAuthnRequest it is a response to	

Returns

a newly created LassoLibAuthnResponse object

Types and Values

struct LassoLibAuthnResponse

```
struct LassoLibAuthnResponse {
  LassoSamlpResponse parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="RelayState" minOccurs="0"/> */
  char *RelayState;
  /* <xs:attribute ref="consent" use="optional"/> */
  char *consent;
};
```

5.9 LassoLibFederationTerminationNotification

LassoLibFederationTerminationNotification — <lib:FederationTerminationNotification>

Functions

LassoNode *	lasso_lib_federation_termination_notification_new ()
LassoNode *	lasso_lib_federation_termination_notification_new_full ()

Types and Values

struct	LassoLibFederationTerminationNotification

Description

```
<xs:element name="FederationTerminationNotification"
    type="FederationTerminationNotificationType"/>
  <xs:complexType name="FederationTerminationNotificationType">
    <xs:complexContent>
      <xs:extension base="samlp:RequestAbstractType">
        <xs:sequence>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="ProviderID"/>
          <xs:element ref="saml:NameIdentifier"/>
        </xs:sequence>
      <xs:attribute ref="consent" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ProviderID" type="md:entityIDType"/>
```

Figure 5.7: Schema fragment for lib:FederationTerminationNotification

Functions

lasso_lib_federation_termination_notification_new ()

```
LassoNode~*
lasso_lib_federation_termination_notification_new
                               (void);
```

Creates a new LassoLibFederationTerminationNotification object.

Returns

a newly created LassoLibFederationTerminationNotification object

lasso_lib_federation_termination_notification_new_full ()

```
LassoNode~*
lasso_lib_federation_termination_notification_new_full
                               (char *providerID,
                                LassoSamlNameIdentifier *nameIdentifier,
                                LassoSignatureType sign_type,
                                LassoSignatureMethod sign_method);
```

Creates a new LassoLibFederationTerminationNotification object and initializes it with the parameters.

Parameters

providerID	the provider ID doing the notification	
nameIdentifier	the name identifier for the federation to terminate.	
sign_type	a LassoSignatureType value	
sign_method	a LassoSignatureMethod value	

Returns

a newly created LassoLibFederationTerminationNotification object

Types and Values

struct LassoLibFederationTerminationNotification

```
struct LassoLibFederationTerminationNotification {
  LassoSamlpRequestAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="saml:NameIdentifier"/> */
  LassoSamlNameIdentifier *NameIdentifier;
  /* <xs:attribute ref="consent" use="optional"/> */
  char *consent;

  char *RelayState;~/* not in schema but allowed in redirects */
};
```

5.10 LassoLibIDPEntries

LassoLibIDPEntries — <lib:IDPEntries>

Functions

LassoNode *	lasso_lib_idp_entries_new ()

Types and Values

struct	LassoLibIDPEntries

Description

```
<xs:element name="IDPEntries">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="IDPEntry" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 5.8: Schema fragment for lib:IDPEntries

Functions

lasso_lib_idp_entries_new ()

```
LassoNode~*
lasso_lib_idp_entries_new (void);
```

Creates a new LassoLibIDPEntries object.

Returns

a newly created LassoLibIDPEntries object

Types and Values

struct LassoLibIDPEntries

```
struct LassoLibIDPEntries {
  LassoNode parent;

  /* <xs:element ref="IDPEntry" maxOccurs="unbounded"/> */
  GList *IDPEntry; /* of LassoLibIDPEntry */
};
```

5.11 LassoLibIDPEntry

LassoLibIDPEntry — <lib:IDPEntry>

Functions

LassoNode *	lasso_lib_idp_entry_new ()

Types and Values

struct	LassoLibIDPEntry

Description

```
<xs:element name="IDPEntry">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ProviderID"/>
      <xs:element name="ProviderName" type="xs:string" minOccurs="0"/>
      <xs:element name="Loc" type="xs:anyURI"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 5.9: Schema fragment for lib:IDPEntry

Functions

lasso_lib_idp_entry_new ()

```
LassoNode~*
lasso_lib_idp_entry_new (void);
```

Creates a new LassoLibIDPEntry object.

Returns

a newly created *LassoLibIDPEntry* object

Types and Values

struct LassoLibIDPEntry

```
struct LassoLibIDPEntry {
  LassoNode parent;

  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element name="ProviderName" type="xs:string" minOccurs="0"/> */
  char *ProviderName;
  /* <xs:element name="Loc" type="xs:anyURI"/> */
  char *Loc;
};
```

5.12 LassoLibIDPList

LassoLibIDPList — <lib:IDPList>

Functions

LassoNode *	lasso_lib_idp_list_new ()

Types and Values

struct	LassoLibIDPList

Description

```
<xs:element name="IDPList" type="IDPListType"/>
<xs:complexType name="IDPListType">
  <xs:sequence>
    <xs:element ref="IDPEntries"/>
    <xs:element ref="GetComplete" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="GetComplete" type="xs:anyURI"/>
```

Figure 5.10: Schema fragment for lib:IDPList

Functions

lasso_lib_idp_list_new ()

```
LassoNode~*
lasso_lib_idp_list_new (void);
```

Creates a new LassoLibIDPList object.

Returns

a newly created LassoLibIDPList object

Types and Values

struct LassoLibIDPList

```
struct LassoLibIDPList {
  LassoNode parent;

  /* <xs:element ref="IDPEntries"/> */
  LassoLibIDPEntries *IDPEntries;
  /* <xs:element ref="GetComplete" minOccurs="0"/> */
  char *GetComplete;
};
```

5.13 LassoLibLogoutRequest

LassoLibLogoutRequest — <lib:LogoutRequest>

Functions

LassoNode *	lasso_lib_logout_request_new ()
LassoNode *	lasso_lib_logout_request_new_full ()

Types and Values

struct	LassoLibLogoutRequest

Description

```
<xs:element name="LogoutRequest" type="LogoutRequestType"/>
<xs:complexType name="LogoutRequestType">
  <xs:complexContent>
    <xs:extension base="samlp:RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ProviderID"/>
        <xs:element ref="saml:NameIdentifier"/>
        <xs:element name="SessionIndex" type="xs:string" minOccurs="0" maxOccurs="unbounded ↩
            "/>
        <xs:element ref="RelayState" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute ref="consent" use="optional"/>
      <xs:attribute name="NotOnOrAfter" type="xs:dateTime" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ProviderID" type="md:entityIDType"/>
<xs:element name="RelayState" type="xs:string"/>
```

Figure 5.11: Schema fragment for lib:LogoutRequest

Functions

lasso_lib_logout_request_new ()

```
LassoNode~*
lasso_lib_logout_request_new (void);
```

Creates a new LassoLibLogoutRequest object.

Returns

a newly created LassoLibLogoutRequest object

lasso_lib_logout_request_new_full ()

```
LassoNode~*
lasso_lib_logout_request_new_full (char *providerID,
                                   LassoSamlNameIdentifier *nameIdentifier,
                                   LassoSignatureType sign_type,
                                   LassoSignatureMethod sign_method);
```

Creates a new LassoLibLogoutRequest object and initializes it with the parameters.

Parameters

providerID	the provider ID requesting the logout	
nameIdentifier	the name identifier to log out	
sign_type	a LassoSignatureType value	
sign_method	a LassoSignatureMethod value	

Returns

a newly created LassoLibLogoutRequest object

Types and Values

struct LassoLibLogoutRequest

```
struct LassoLibLogoutRequest {
  LassoSamlpRequestAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  char *ProviderID;
  LassoSamlNameIdentifier *NameIdentifier;
  char *SessionIndex;
  char *RelayState;
  char *consent;
  char *NotOnOrAfter;
};
```

5.14 LassoLibLogoutResponse

LassoLibLogoutResponse — <lib:LogoutResponse>

Functions

LassoNode *	lasso_lib_logout_response_new ()
LassoNode *	lasso_lib_logout_response_new_full ()

Types and Values

struct	LassoLibLogoutResponse

Description

```
<xs:element name="LogoutResponse" type="StatusResponseType"/>
```

Figure 5.12: Schema fragment for lib:LogoutResponse

Functions

lasso_lib_logout_response_new ()

```
LassoNode~*
lasso_lib_logout_response_new (void);
```

Creates a new LassoLibLogoutResponse object.

Returns

a newly created LassoLibLogoutResponse object

lasso_lib_logout_response_new_full ()

```
LassoNode~*
lasso_lib_logout_response_new_full (char *providerID,
                                    const char *statusCodeValue,
                                    LassoLibLogoutRequest *request,
                                    LassoSignatureType sign_type,
                                    LassoSignatureMethod sign_method);
```

Creates a new LassoLibLogoutResponse object and initializes it with the parameters.

Parameters

providerID	the providerID of the responded	
statusCodeValue	a response status code	
request	the request this is a response to	
sign_type	a LassoSignatureType value	
sign_method	a LassoSignatureMethod value	

Returns

a newly created LassoLibLogoutResponse object

Types and Values

struct LassoLibLogoutResponse

```
struct LassoLibLogoutResponse {
  LassoLibStatusResponse parent;
};
```

5.15 LassoLibNameIdentifierMappingRequest

LassoLibNameIdentifierMappingRequest — <lib:NameIdentifierMappingRequest>

Functions

LassoNode *	lasso_lib_name_identifier_mapping_request_new ()
LassoNode *	lasso_lib_name_identifier_mapping_request_new_full ()

Types and Values

struct	LassoLibNameIdentifierMappingRequest

Description

```
<xs:element name="NameIdentifierMappingRequest" type="NameIdentifierMappingRequestType"/>
<xs:complexType name="NameIdentifierMappingRequestType">
  <xs:complexContent>
    <xs:extension base="samlp:RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ProviderID"/>
        <xs:element ref="saml:NameIdentifier"/>
        <xs:element name="TargetNamespace" type="md:entityIDType"/>
      </xs:sequence>
      <xs:attribute ref="consent" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ProviderID" type="md:entityIDType"/>
```

Figure 5.13: Schema fragment for lib:NameIdentifierMappingRequest

Functions

lasso_lib_name_identifier_mapping_request_new ()

```
LassoNode~*
lasso_lib_name_identifier_mapping_request_new
                                (void);
```

Creates a new LassoLibNameIdentifierMappingRequest object.

Returns

a newly created LassoLibNameIdentifierMappingRequest object

lasso_lib_name_identifier_mapping_request_new_full ()

```
LassoNode~*
lasso_lib_name_identifier_mapping_request_new_full
                                (char *providerID,
                                 LassoSamlNameIdentifier *nameIdentifier,
                                 const char *targetNamespace,
                                 LassoSignatureType sign_type,
                                 LassoSignatureMethod sign_method);
```

Creates a new LassoLibNameIdentifierMappingRequest object and initializes it with the parameters. It also setups the signature on the request object, you must preceise the signing key later.

Parameters

providerID	the provider ID requesting the name identifier mapping	
nameIdentifier	a LassoSamlNameIdentifier object	
targetNamespace	an URI for the target namespace	
sign_type	a LassoSignatureType value	
sign_method	a LassoSignatureMethod value	

Returns

a newly created LassoLibNameIdentifierMappingRequest object

Types and Values

struct LassoLibNameIdentifierMappingRequest

```
struct LassoLibNameIdentifierMappingRequest {
  LassoSamlpRequestAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="saml:NameIdentifier"/> */
  LassoSamlNameIdentifier *NameIdentifier;
  /* <xs:element name="TargetNamespace" type="md:entityIDType"/> */
  char *TargetNamespace;
  /* <xs:attribute ref="consent" use="optional"/> */
  char *consent;
};
```

5.16 LassoLibNameIdentifierMappingResponse

LassoLibNameIdentifierMappingResponse — <lib:NameIdentifierMappingResponse>

Functions

LassoNode *	lasso_lib_name_identifier_mapping_response_new ()
LassoNode *	lasso_lib_name_identifier_mapping_response_new_full ()

Types and Values

struct	LassoLibNameIdentifierMappingResponse

Description

```
<xs:element name="NameIdentifierMappingResponse" type="NameIdentifierMappingResponseType"/>
<xs:complexType name="NameIdentifierMappingResponseType">
  <xs:complexContent>
    <xs:extension base="samlp:ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ProviderID"/>
        <xs:element ref="samlp:Status"/>
        <xs:element ref="saml:NameIdentifier" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 5.14: Schema fragment for lib:NameIdentifierMappingResponse

Functions

lasso_lib_name_identifier_mapping_response_new ()

```
LassoNode~*
lasso_lib_name_identifier_mapping_response_new
                                (void);
```

Creates a new LassoLibNameIdentifierMappingResponse object.

Returns

a newly created LassoLibNameIdentifierMappingResponse object

lasso_lib_name_identifier_mapping_response_new_full ()

```
LassoNode~*
lasso_lib_name_identifier_mapping_response_new_full
                               (char *provideRID,
                                const char *statusCodeValue,
                                LassoLibNameIdentifierMappingRequest *request,
                                LassoSignatureType sign_type,
                                LassoSignatureMethod sign_method);
```

Creates a new LassoLibNameIdentifierMappingResponse object and initializes it with the parameters.

Parameters

providerID	the providerID of the responder	
statusCodeValue	a response status code	
request	the request which is asnwered by this response	
sign_type	a LassoSignatureType value	
sign_method	a LassoSignatureMethod value	

Returns

a newly created LassoLibNameIdentifierMappingResponse object

Types and Values

struct LassoLibNameIdentifierMappingResponse

```
struct LassoLibNameIdentifierMappingResponse {
  LassoSamlpResponseAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="samlp:Status"/> */
  LassoSamlpStatus *Status;
  /* <xs:element ref="saml:NameIdentifier" minOccurs="0"/> */
  LassoSamlNameIdentifier *NameIdentifier;
};
```

5.17 LassoLibRegisterNameIdentifierRequest

LassoLibRegisterNameIdentifierRequest — <lib:RegisterNameIdentifierRequest>

Functions

LassoNode *	lasso_lib_register_name_identifier_request_new ()
LassoNode *	lasso_lib_register_name_identifier_request_new_full ()

Types and Values

struct	LassoLibRegisterNameIdentifierRequest

Description

```
<xs:element name="RegisterNameIdentifierRequest" type="RegisterNameIdentifierRequestType"/>
<xs:complexType name="RegisterNameIdentifierRequestType">
  <xs:complexContent>
    <xs:extension base="samlp:RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ProviderID"/>
        <xs:element ref="IDPProvidedNameIdentifier"/>
        <xs:element ref="SPProvidedNameIdentifier" minOccurs="0"/>
        <xs:element ref="OldProvidedNameIdentifier"/>
        <xs:element ref="RelayState" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="IDPProvidedNameIdentifier" type="saml:NameIdentifierType"/>
<xs:element name="SPProvidedNameIdentifier" type="saml:NameIdentifierType"/>
<xs:element name="OldProvidedNameIdentifier" type="saml:NameIdentifierType"/>

<xs:element name="ProviderID" type="md:entityIDType"/>
<xs:element name="RelayState" type="xs:string"/>
```

Figure 5.15: Schema fragment for lib:RegisterNameIdentifierRequest

Functions

lasso_lib_register_name_identifier_request_new ()

```
LassoNode~*
lasso_lib_register_name_identifier_request_new
                               (void);
```

Creates a new LassoLibRegisterNameIdentifierRequest object.

Returns

a newly created LassoLibRegisterNameIdentifierRequest object

lasso_lib_register_name_identifier_request_new_full ()

```
LassoNode~*
lasso_lib_register_name_identifier_request_new_full
                               (const char *providerID,
                                LassoSamlNameIdentifier *idpNameIdentifier,
                                LassoSamlNameIdentifier *spNameIdentifier,
                                LassoSamlNameIdentifier *oldNameIdentifier,
                                LassoSignatureType sign_type,
                                LassoSignatureMethod sign_method);
```

Creates a new LassoLibRegisterNameIdentifierRequest object and initializes it with the parameters.

Parameters

providerID	the providerID of the requester
idpNameIdentifier	a LassoSamlNameIdentifier object, giving the new idp provided name identifier
spNameIdentifier	a LassoSamlNameIdentifier object, giving the new sp provided name identifier
oldNameIdentifier	a LassoSamlNameIdentifier object, giving the old name identifier
sign_type	a LassoSignatureType value
sign_method	a LassoSignatureMethod value

Returns

a newly created LassoLibRegisterNameIdentifierRequest object

Types and Values

struct LassoLibRegisterNameIdentifierRequest

```
struct LassoLibRegisterNameIdentifierRequest {
  LassoSamlpRequestAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="IDPProvidedNameIdentifier"/> */
  LassoSamlNameIdentifier *IDPProvidedNameIdentifier;
  /* <xs:element ref="SPProvidedNameIdentifier" minOccurs="0"/> */
  LassoSamlNameIdentifier *SPProvidedNameIdentifier;
  /* <xs:element ref="OldProvidedNameIdentifier"/> */
  LassoSamlNameIdentifier *OldProvidedNameIdentifier;
  /* <xs:element ref="RelayState" minOccurs="0"/> */
  char *RelayState;
};
```

5.18 LassoLibRegisterNameIdentifierResponse

LassoLibRegisterNameIdentifierResponse — <lib:RegisterNameIdentifierResponse>

Functions

LassoNode *	lasso_lib_register_name_identifier_response_new ()
LassoNode *	lasso_lib_register_name_identifier_response_new_full ()

Types and Values

struct	LassoLibRegisterNameIdentifierResponse

Description

```
<xs:element name="RegisterNameIdentifierResponse" type="StatusResponseType"/>
```

Figure 5.16: Schema fragment for lib:RegisterNameIdentifierResponse

Functions

lasso_lib_register_name_identifier_response_new ()

```
LassoNode~*
lasso_lib_register_name_identifier_response_new
                                (void);
```

Creates a new LassoLibRegisterNameIdentifierResponse object.

Returns

a newly created LassoLibRegisterNameIdentifierResponse object

lasso_lib_register_name_identifier_response_new_full ()

```
LassoNode~*
lasso_lib_register_name_identifier_response_new_full
                                (const char *providerID,
                                 const char *statusCodeValue,
                                 LassoLibRegisterNameIdentifierRequest *request,
                                 LassoSignatureType sign_type,
                                 LassoSignatureMethod sign_method);
```

Creates a new LassoLibRegisterNameIdentifierResponse object and initializes it with the parameters.

Parameters

providerID	the providerID of the responder	
statusCodeValue	a response status code	
request	the request which is answered by this response	
sign_type	a LassoSignatureType value	
sign_method	a LassoSignatureMethod value	

Returns

a newly created LassoLibRegisterNameIdentifierResponse object

Types and Values

struct LassoLibRegisterNameIdentifierResponse

```
struct LassoLibRegisterNameIdentifierResponse {
  LassoLibStatusResponse parent;
};
```

5.19 LassoLibRequestAuthnContext

LassoLibRequestAuthnContext — <lib:RequestAuthnContext>

Functions

LassoLibRequestAuthnContext *	lasso_lib_request_authn_context_new ()

Types and Values

struct	LassoLibRequestAuthnContext

Description

Information describing which authentication context the requester desires the identity provider to use in authenticating the Principal.

```
<xs:element name="RequestAuthnContext">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element name="AuthnContextClassRef" type="xs:anyURI" maxOccurs="unbounded"/>
        <xs:element name="AuthnContextStatementRef" type="xs:anyURI" maxOccurs="unbounded ↩
            "/>
      </xs:choice>
      <xs:element name="AuthnContextComparison"
          type="AuthnContextComparisonType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 5.17: Schema fragment for lib:RequestAuthnContext

Functions

lasso_lib_request_authn_context_new ()

```
LassoLibRequestAuthnContext~*
lasso_lib_request_authn_context_new (void);
```

Creates a new LassoLibRequestAuthnContext object.

Returns

a newly created LassoLibRequestAuthnContext object

Types and Values

struct LassoLibRequestAuthnContext

```
struct LassoLibRequestAuthnContext {
  LassoNode parent;

  /* <xs:element name="AuthnContextClassRef" type="xs:anyURI" maxOccurs="unbounded"/> */
  GList *AuthnContextClassRef; /* of strings */
  /* <xs:element name="AuthnContextStatementRef" type="xs:anyURI" maxOccurs="unbounded"/>  ↩
      */
  GList *AuthnContextStatementRef; /* of strings */
  /* <xs:element name="AuthnContextComparison" type="AuthnContextComparisonType"
   *   minOccurs="0"/> */
  char *AuthnContextComparison;
};
```

5.20 LassoLibScoping

LassoLibScoping — <lib:Scoping>

Functions

LassoLibScoping *	lasso_lib_scoping_new ()

Types and Values

struct	LassoLibScoping

Description

```
<xs:complexType name="ScopingType">
  <xs:sequence>
    <xs:element name="ProxyCount" type="xs:nonNegativeInteger" minOccurs="0"/>
    <xs:element ref="IDPList" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Scoping" type="ScopingType"/>
```

Figure 5.18: Schema fragment for lib:Scoping

Functions

lasso_lib_scoping_new ()

```
LassoLibScoping~*
lasso_lib_scoping_new (void);
```

Creates a new LassoLibScoping object.

Returns

a newly created LassoLibScoping object

Types and Values

struct LassoLibScoping

```
struct LassoLibScoping {
  LassoNode parent;

  /* <xs:element name="ProxyCount" type="xs:nonNegativeInteger" minOccurs="0"/> */
  int ProxyCount;
  /* <xs:element ref="IDPList" minOccurs="0"/> */
  LassoLibIDPList *IDPList;
};
```

5.21 LassoLibStatusResponse

LassoLibStatusResponse — <lib:StatusResponse>

Functions

LassoNode *	lasso_lib_status_response_new ()

Types and Values

struct	LassoLibStatusResponse

Description

```
<xs:complexType name="StatusResponseType">
  <xs:complexContent>
    <xs:extension base="samlp:ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ProviderID"/>
        <xs:element ref="samlp:Status"/>
        <xs:element ref="RelayState" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ProviderID" type="md:entityIDType"/>
<xs:element name="RelayState" type="xs:string"/>
```

Figure 5.19: Schema fragment for lib:StatusResponse

Functions

lasso_lib_status_response_new ()

```
LassoNode~*
lasso_lib_status_response_new (void);
```

Creates a new LassoLibStatusResponse object.

Returns

a newly created LassoLibStatusResponse object

Types and Values

struct LassoLibStatusResponse

```
struct LassoLibStatusResponse {
  LassoSamlpResponseAbstract parent;

  /* <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Extension; /* of xmlNode* */
  /* <xs:element ref="ProviderID"/> */
  char *ProviderID;
  /* <xs:element ref="samlp:Status"/> */
  LassoSamlpStatus *Status;
  /* <xs:element ref="RelayState" minOccurs="0"/> */
  char *RelayState;
};
```

5.22 LassoLibSubject

LassoLibSubject — <lib:Subject>

Functions

LassoLibSubject *	lasso_lib_subject_new ()

Types and Values

struct	LassoLibSubject

Description

```
<xs:complexType name="SubjectType">
  <xs:complexContent>
    <xs:extension base="saml:SubjectType">
      <xs:sequence>
        <xs:element ref="IDPProvidedNameIdentifier"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="Subject" type="SubjectType" substitutionGroup="saml:Subject"/>
```

Figure 5.20: Schema fragment for lib:Subject

Functions

lasso_lib_subject_new ()

```
LassoLibSubject~*
lasso_lib_subject_new (void);
```

Creates a new LassoLibSubject object.

Returns

a newly created LassoLibSubject object

Types and Values

struct LassoLibSubject

```
struct LassoLibSubject {
  LassoSamlSubject parent;

  /* <xs:element ref="IDPProvidedNameIdentifier"/> */
  LassoSamlNameIdentifier *IDPProvidedNameIdentifier;
};
```

5.23 LassoPaosRequest

LassoPaosRequest — <paos:Request>

Functions

int	lasso_paos_request_validate ()
LassoNode *	lasso_paos_request_new ()

Description

```
<xs:element name="Request" type="RequestType"/>
<xs:complexType name="RequestType">
    <xs:attribute name="responseConsumerURL" type="xs:anyURI" use="required"/>
    <xs:attribute name="service" type="xs:anyURI" use="required"/>
    <xs:attribute name="messageID" type="IDType" use="optional"/>
    <xs:attribute ref="S:mustUnderstand" use="required"/>
    <xs:attribute ref="S:actor" use="required"/>
</xs:complexType>
```

Figure 5.21: Schema fragment for paos:Request

Functions

lasso_paos_request_validate ()

```
int
lasso_paos_request_validate (LassoPaosRequest *node);
```

Validates the object conforms to required values.

- responseConsumerURL must be non-NULL
- mustUnderstand must be TRUE
- actor must be equal to LASSO_SOAP_ENV_ACTOR
- service must be equal to LASSO_ECP_HREF

Parameters

request	The LassoPaosRequest object to validate

Returns

0 on success, error code otherwise

lasso_paos_request_new ()

```
LassoNode~*
lasso_paos_request_new (const gchar *responseConsumerURL,
                        const gchar *messageID);
```

The LassoPaosRequest object is initialized as follows:

```
responseConsumerURL = responseConsumerURL (if non-NULL)
messageID = messageID (if non-NULL) otherwise generated unique id
mustUnderstand = TRUE
actor = LASSO_SOAP_ENV_ACTOR
service = LASSO_ECP_HREF
```

Parameters

responseConsumerURL	.	*[allow-none]*
messageID	.	*[allow-none]*

Returns

newly created & initialized LassoPaosRequest object

5.24 LassoPaosResponse

LassoPaosResponse — <paos:Response>

Functions

int	lasso_paos_response_validate ()
LassoNode *	lasso_paos_response_new ()

Description

```
<xs:element name="Response" type="ResponseType"/>
<xs:complexType name="ResponseType">
    <xs:attribute name="refToMessageID" type="IDType" use="optional"/>
    <xs:attribute ref="S:mustUnderstand" use="required"/>
    <xs:attribute ref="S:actor" use="required"/>
</xs:complexType>
```

Figure 5.22: Schema fragment for paos:Response

Functions

lasso_paos_response_validate ()

```
int
lasso_paos_response_validate (LassoPaosResponse *node);
```

Validates the object conforms to required values.

- mustUnderstand must be TRUE
- actor must be equal to LASSO_SOAP_ENV_ACTOR

Parameters

response	The LassoPaosResponse object to validate	

Returns

0 on success, error code otherwise

lasso_paos_response_new ()

```
LassoNode~*
lasso_paos_response_new (const gchar *refToMessageID);
```

The LassoPaosResponse object is initialized as follows:

```
refToMessageID = refToMessageID (if non-NULL)
mustUnderstand = TRUE
actor = LASSO_SOAP_ENV_ACTOR
```

Parameters

refToMessageID	.	*[allow-none]*

Returns

a newly created and initialized LassoPaosResponse object

5.25 LassoSamlAdvice

LassoSamlAdvice — <saml:Advice>

Functions

LassoNode *	lasso_saml_advice_new ()

Types and Values

struct	LassoSamlAdvice

Description

```
<element name="Advice" type="saml:AdviceType"/>
<complexType name="AdviceType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="saml:AssertionIDReference"/>
    <element ref="saml:Assertion"/>
    <any namespace="##other" processContents="lax"/>
  </choice>
</complexType>

<element name="AssertionIDReference" type="saml:IDReferenceType"/>
<simpleType name="IDReferenceType">
  <restriction base="string"/>
</simpleType>
```

Figure 5.23: Schema fragment for saml:Advice

Functions

lasso_saml_advice_new ()

```
LassoNode~*
lasso_saml_advice_new (void);
```

Creates a new LassoSamlAdvice object.

Returns

a newly created LassoSamlAdvice

Types and Values

struct LassoSamlAdvice

```
struct LassoSamlAdvice {
  LassoNode parent;

  /* <element ref="saml:AssertionIDReference"/> */
  GList *AssertionIDReference; /* of LassoNode */
  /* <element ref="saml:Assertion"/> */
  struct _LassoSamlAssertion *Assertion;
};
```

5.26 LassoSamlAssertion

LassoSamlAssertion — <saml:Assertion>

Functions

LassoSamlAssertion *	lasso_saml_assertion_new ()

Types and Values

struct	LassoSamlAssertion

Description

```
<element name="Assertion" type="saml:AssertionType"/>
<complexType name="AssertionType">
  <sequence>
    <element ref="saml:Conditions" minOccurs="0"/>
    <element ref="saml:Advice" minOccurs="0"/>
    <choice maxOccurs="unbounded">
      <element ref="saml:Statement"/>
      <element ref="saml:SubjectStatement"/>
      <element ref="saml:AuthenticationStatement"/>
      <element ref="saml:AuthorizationDecisionStatement"/>
      <element ref="saml:AttributeStatement"/>
    </choice>
    <element ref="ds:Signature" minOccurs="0"/>
  </sequence>
  <attribute name="MajorVersion" type="integer" use="required"/>
  <attribute name="MinorVersion" type="integer" use="required"/>
  <attribute name="AssertionID" type="saml:IDType" use="required"/>
  <attribute name="Issuer" type="string" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
</complexType>

From oasis-sstc-saml-schema-assertion-1.0.xsd:
<simpleType name="IDType">
  <restriction base="string"/>
</simpleType>
```

Figure 5.24: Schema fragment for saml:Assertion

Functions

lasso_saml_assertion_new ()

```
LassoSamlAssertion~*
lasso_saml_assertion_new (void);
```

Creates a new LassoSamlAssertion object.

Returns

a newly created LassoSamlAssertion object

Types and Values

struct LassoSamlAssertion

```
struct LassoSamlAssertion {
  LassoNode parent;

  /* <element ref="saml:Conditions" minOccurs="0"/> */
  LassoSamlConditions *Conditions;
  /* <element ref="saml:Advice" minOccurs="0"/> */
  LassoSamlAdvice *Advice;
  void *Statement; /* XXX LassoSamlStatement missing from lasso */
  LassoSamlSubjectStatement *SubjectStatement;
  LassoSamlAuthenticationStatement *AuthenticationStatement;
  void *AuthorizationDecisionStatement;
    /* XXX LassoSamlAuthorizationDecisionStatement missing from lasso*/
  LassoSamlAttributeStatement *AttributeStatement;

  int MajorVersion;
  int MinorVersion;
  char *AssertionID;
  char *Issuer;
  char *IssueInstant;

  /* ds:Signature stuff */
  LassoSignatureType sign_type;
  LassoSignatureMethod sign_method;
  char *private_key_file;
  char *certificate_file;
};
```

5.27 LassoSamlAttributeDesignator

LassoSamlAttributeDesignator — object mapping for a saml:AttributeDesignator

Functions

LassoNode *	lasso_saml_attribute_designator_new ()

Types and Values

struct	LassoSamlAttributeDesignator

Description

The schema fragment (oasis-sstc-saml-schema-assertion-1.1.xsd) is:

```
<element name="AttributeDesignator" type="saml:AttributeDesignatorType"/>
<complexType name="AttributeDesignatorType">
  <attribute name="AttributeName" type="string" use="required"/>
  <attribute name="AttributeNamespace" type="anyURI" use="required"/>
</complexType>
```

Figure 5.25: Schema fragment for saml:AttributeDesignator

Functions

lasso_saml_attribute_designator_new ()

```
LassoNode~*
lasso_saml_attribute_designator_new (void);
```

Creates a new LassoSamlAttributeDesignator object.

Returns

a newly created LassoSamlAttributeDesignator object

Types and Values

struct LassoSamlAttributeDesignator

```
struct LassoSamlAttributeDesignator {
  LassoNode parent;

  char *AttributeName;
  char *AttributeNamespace;
};
```

5.28 LassoSamlAttributeStatement

LassoSamlAttributeStatement — object mapping for a saml:AttributeStatement

Functions

LassoSamlAttributeStatement *	lasso_saml_attribute_statement_new ()

Types and Values

struct	LassoSamlAttributeStatement

Description

The schema fragment (oasis-sstc-saml-schema-assertion-1.1.xsd):

```
<element name="AttributeStatement" type="saml:AttributeStatementType"/>
<complexType name="AttributeStatementType">
  <complexContent>
    <extension base="saml:SubjectStatementAbstractType">
      <sequence>
        <element ref="saml:Attribute" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 5.26: Schema fragment for saml:AttributeStatement

Functions

lasso_saml_attribute_statement_new ()

```
LassoSamlAttributeStatement~*
lasso_saml_attribute_statement_new (void);
```

Creates a new LassoSamlAttributeStatement object.

Returns

a newly created LassoSamlAttributeStatement object

Types and Values

struct LassoSamlAttributeStatement

```
struct LassoSamlAttributeStatement {
  LassoSamlSubjectStatementAbstract parent;

  GList *Attribute; /* of LassoNode */
};
```

5.29 LassoSamlAttributeValue

LassoSamlAttributeValue — value of an attribute in SAML 1.0/1.1 assertion

Functions

LassoSamlAttributeValue *	lasso_saml_attribute_value_new ()

Types and Values

struct	LassoSamlAttributeValue

Includes

```
#include <lasso/xml/saml_attribute_value.h>
```

Description

```
<element name="AttributeValue" type="anyType"/>
```

Figure 5.27: Schema fragment for samlp2:Extensions

This object support a special of specifying its content. If the `any` field is NULL, then you can attach an xmlNode using lasso_node_set_original_xmlnode() and it will be used to generate the content of the serialized xmlNode for this object. The content (attributes, childrent and namespaces) of the xmlNode will be copied to the result node returned by calls to

lasso_node_get_xmlNode().

Functions

lasso_saml_attribute_value_new ()

```
LassoSamlAttributeValue~*
lasso_saml_attribute_value_new (void);
```

Types and Values

struct LassoSamlAttributeValue

```
struct LassoSamlAttributeValue {
  LassoNode parent;

  GList *any; /* of LassoNode */
};
```

See Also

LassoSamlAttribute, LassoSamlAttributeStatement, LassoSamlAssertion

5.30 LassoSamlAttribute

LassoSamlAttribute —

Functions

LassoSamlAttribute *	lasso_saml_attribute_new ()

Types and Values

struct	LassoSamlAttribute

Description

Functions

lasso_saml_attribute_new ()

```
LassoSamlAttribute~*
lasso_saml_attribute_new (void);
```

Creates a new LassoSamlAttribute object.

Returns

a newly created LassoSamlAttribute object

Types and Values

struct LassoSamlAttribute

```
struct LassoSamlAttribute {
  LassoSamlAttributeDesignator parent;

  gchar *attributeName;
  gchar *attributeNameSpace;
  GList *AttributeValue; /* of LassoNode */
};
```

5.31 LassoSamlAudienceRestrictionCondition

LassoSamlAudienceRestrictionCondition —

Functions

LassoSamlAudienceRestrictionCondition *	lasso_saml_audience_restriction_condition_new ()
LassoSamlAudienceRestrictionCondition *	lasso_saml_audience_restriction_condition_new_full ()

Types and Values

struct	LassoSamlAudienceRestrictionCondition

Description

Functions

lasso_saml_audience_restriction_condition_new ()

```
LassoSamlAudienceRestrictionCondition~*
lasso_saml_audience_restriction_condition_new
                               (void);
```

Creates a new LassoSamlAudienceRestrictionCondition object.

Returns

a newly created LassoSamlAudienceRestrictionCondition

lasso_saml_audience_restriction_condition_new_full ()

```
LassoSamlAudienceRestrictionCondition~*
lasso_saml_audience_restriction_condition_new_full
                               (const char *audience);
```

Creates a new LassoSamlAudienceRestrictionCondition object and initializes it with the parameters.

Parameters

audience	a string which specify to which audience the restriction condition applies

Returns

a newly created LassoSamlAudienceRestrictionCondition

Types and Values

struct LassoSamlAudienceRestrictionCondition

```
struct LassoSamlAudienceRestrictionCondition {
  LassoSamlConditionAbstract parent;

  /* <element ref="saml:Audience" maxOccurs="unbounded"/> */
  GList *Audience; /* of strings */
};
```

5.32 LassoSamlAuthenticationStatement

LassoSamlAuthenticationStatement — <saml:AuthenticationStatement>

Functions

LassoNode *	lasso_saml_authentication_statement_new ()

Types and Values

struct	LassoSamlAuthenticationStatement

Description

```
<element name="AuthenticationStatement" type="saml:AuthenticationStatementType"/>
<complexType name="AuthenticationStatementType">
  <complexContent>
    <extension base="saml:SubjectStatementAbstractType">
      <sequence>
        <element ref="saml:SubjectLocality" minOccurs="0"/>
        <element ref="saml:AuthorityBinding" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="AuthenticationMethod" type="anyURI" use="required"/>
      <attribute name="AuthenticationInstant" type="dateTime" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 5.28: Schema fragment for saml:AuthenticationStatement

Functions

lasso_saml_authentication_statement_new ()

```
LassoNode~*
lasso_saml_authentication_statement_new
                                (void);
```

Creates a new LassoSamlAuthenticationStatement object.

Returns

a newly created LassoSamlAuthenticationStatement object

Types and Values

struct LassoSamlAuthenticationStatement

```
struct LassoSamlAuthenticationStatement {
  LassoSamlSubjectStatementAbstract parent;

  /* <element ref="saml:SubjectLocality" minOccurs="0"/> */
  LassoSamlSubjectLocality *SubjectLocality;
  /* <element ref="saml:AuthorityBinding" minOccurs="0" maxOccurs="unbounded"/> */
  GList *AuthorityBinding; /* of LassoNode */
  /* <attribute name="AuthenticationMethod" type="anyURI" use="required"/> */
  char *AuthenticationMethod;
  /* <attribute name="AuthenticationInstant" type="dateTime" use="required"/> */
  char *AuthenticationInstant;
};
```

5.33 LassoSamlAuthorityBinding

LassoSamlAuthorityBinding — <saml:AuthorityBinding>

Functions

LassoNode *	lasso_saml_authority_binding_new ()

Types and Values

struct	LassoSamlAuthorityBinding

Description

```
<element name="AuthorityBinding" type="saml:AuthorityBindingType"/>
<complexType name="AuthorityBindingType">
  <attribute name="AuthorityKind" type="QName" use="required"/>
  <attribute name="Location" type="anyURI" use="required"/>
  <attribute name="Binding" type="anyURI" use="required"/>
</complexType>
```

Figure 5.29: Schema fragment for saml:AuthorityBinding

Functions

lasso_saml_authority_binding_new ()

```
LassoNode~*
lasso_saml_authority_binding_new (void);
```

Creates a new LassoSamlAuthorityBinding object.

Returns

a newly created LassoSamlAuthorityBinding object

Types and Values

struct LassoSamlAuthorityBinding

```
struct LassoSamlAuthorityBinding {
  LassoNode parent;

  /* <attribute name="AuthorityKind" type="QName" use="required"/> */
  char *AuthorityKind;
  /* <attribute name="Location" type="anyURI" use="required"/> */
  char *Location;
  /* <attribute name="Binding" type="anyURI" use="required"/> */
  char *Binding;
};
```

5.34 LassoSamlConditionAbstract

LassoSamlConditionAbstract — <:saml:ConditionAbstract>

Types and Values

struct	LassoSamlConditionAbstract

Description

```
<element name="Condition" type="saml:ConditionAbstractType"/>
<complexType name="ConditionAbstractType" abstract="true"/>
```

Figure 5.30: Schema fragment for saml:ConditionAbstract

Functions

Types and Values

struct LassoSamlConditionAbstract

```
struct LassoSamlConditionAbstract {
  LassoNode parent;
};
```

5.35 LassoSamlConditions

LassoSamlConditions — <saml:Conditions>

Functions

LassoSamlConditions *	lasso_saml_conditions_new ()

Types and Values

struct	LassoSamlConditions

Description

```
<element name="Conditions" type="saml:ConditionsType"/>
<complexType name="ConditionsType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="saml:AudienceRestrictionCondition"/>
    <element ref="saml:Condition"/>
  </choice>
  <attribute name="NotBefore" type="dateTime" use="optional"/>
  <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
</complexType>
```

Figure 5.31: Schema fragment for saml:Conditions

Functions

lasso_saml_conditions_new ()

```
LassoSamlConditions~*
lasso_saml_conditions_new (void);
```

Creates a new LassoSamlConditions object.

Returns

a newly created LassoSamlConditions object

Types and Values

struct LassoSamlConditions

```
struct LassoSamlConditions {
  LassoNode parent;

  /* <element ref="saml:Condition"/> */
  GList *Condition; /* of LassoNode */
  /* <element ref="saml:AudienceRestrictionCondition"/> */
  GList *AudienceRestrictionCondition; /* of LassoSamlAudienceRestrictionCondition */
  /* <attribute name="NotBefore" type="dateTime" use="optional"/> */
  char *NotBefore;
  /* <attribute name="NotOnOrAfter" type="dateTime" use="optional"/> */
  char *NotOnOrAfter;
};
```

5.36 LassoSamlNameIdentifier

LassoSamlNameIdentifier — <saml:NameIdentifier>

Functions

LassoSamlNameIdentifier *	lasso_saml_name_identifier_new ()
LassoSamlNameIdentifier *	lasso_saml_name_identifier_new_from_xmlNode ()

Types and Values

struct	LassoSamlNameIdentifier

Description

```
<element name="NameIdentifier" type="saml:NameIdentifierType"/>
<complexType name="NameIdentifierType">
  <simpleContent>
    <extension base="string">
      <attribute name="NameQualifier" type="string" use="optional"/>
      <attribute name="Format" type="anyURI" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

Figure 5.32: Schema fragment for saml:NameIdentifier

Functions

lasso_saml_name_identifier_new ()

```
LassoSamlNameIdentifier~*
lasso_saml_name_identifier_new (void);
```

Creates a new LassoSamlNameIdentifier object.

Returns

a newly created LassoSamlNameIdentifier object

lasso_saml_name_identifier_new_from_xmlNode ()

```
LassoSamlNameIdentifier~*
lasso_saml_name_identifier_new_from_xmlNode
                                (xmlNode *xmlnode);
```

Types and Values

struct LassoSamlNameIdentifier

```
struct LassoSamlNameIdentifier {
  LassoNode parent;

  char *NameQualifier;
  char *Format;
  char *content;
};
```

NameQualifier is the provider ID of the provider that created the name identifier.

Format is a string constant defined by the Liberty Alliance. The following constants are defined: LASSO_LIB_NAME_IDENTIFIER_F LASSO_LIB_NAME_IDENTIFIER_FORMAT_ONE_TIME, LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENCRYPTED (when providers transmit name identifiers) and LASSO_LIB_NAME_IDENTIFIER_FORMAT_ENTITYID.

5.37 LassoSamlpRequestAbstract

LassoSamlpRequestAbstract — <samlp:RequestAbstractType>

Types and Values

struct	LassoSamlpRequestAbstract

Description

```
<complexType name="RequestAbstractType" abstract="true">
  <sequence>
    <element ref="samlp:RespondWith" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="ds:Signature" minOccurs="0"/>
  </sequence>
  <attribute name="RequestID" type="saml:IDType" use="required"/>
  <attribute name="MajorVersion" type="integer" use="required"/>
  <attribute name="MinorVersion" type="integer" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
</complexType>

<element name="RespondWith" type="QName"/>

From oasis-sstc-saml-schema-assertion-1.0.xsd:
<simpleType name="IDType">
  <restriction base="string"/>
</simpleType>
```

Figure 5.33: Schema fragment for samlp:RequestAbstractType

Functions

Types and Values

struct LassoSamlpRequestAbstract

```
struct LassoSamlpRequestAbstract {
  LassoNode parent;

  /* <element ref="samlp:RespondWith" minOccurs="0" maxOccurs="unbounded"/> */
  GList *RespondWith; /* of strings */
  /* <attribute name="RequestID" type="saml:IDType" use="required"/> */
  char *RequestID;
  /* <attribute name="MajorVersion" type="integer" use="required"/> */
  int MajorVersion;
  /* <attribute name="MinorVersion" type="integer" use="required"/> */
  int MinorVersion;
  /* <attribute name="IssueInstant" type="dateTime" use="required"/> */
  char *IssueInstant;

  /* ds:Signature stuffs */
  LassoSignatureType sign_type;
  LassoSignatureMethod sign_method;
  char *private_key_file;
  char *certificate_file;
};
```

5.38 LassoSamlpRequest

LassoSamlpRequest — <samlp:Request>

Functions

LassoNode *	lasso_samlp_request_new ()

Types and Values

struct	LassoSamlpRequest

Description

```
<element name="Request" type="samlp:RequestType"/>
<complexType name="RequestType">
   <complexContent>
     <extension base="samlp:RequestAbstractType">
  <choice>
     <element ref="samlp:Query"/>
     <element ref="samlp:SubjectQuery"/>
     <element ref="samlp:AuthenticationQuery"/>
     <element ref="samlp:AttributeQuery"/>
     <element ref="samlp:AuthorizationDecisionQuery"/>
     <element ref="saml:AssertionIDReference" maxOccurs="unbounded"/>
     <element ref="samlp:AssertionArtifact" maxOccurs="unbounded"/>
  </choice>
     </extension>
   </complexContent>
</complexType>
```

Figure 5.34: Schema fragment for samlp:Request

Functions

lasso_samlp_request_new ()

```
LassoNode~*
lasso_samlp_request_new (void);
```

Creates a new LassoSamlpRequest object.

Returns

a newly created LassoSamlpRequest object

Types and Values

struct LassoSamlpRequest

```
struct LassoSamlpRequest {
  LassoSamlpRequestAbstract parent;

  /* <element name="AssertionArtifact" type="string"/> */
  char *AssertionArtifact;
};
```

5.39 LassoSamlpResponseAbstract

LassoSamlpResponseAbstract — <samlp:ResponseAbstractType>

Functions

void	lasso_samlp_response_abstract_fill ()

Types and Values

struct	LassoSamlpResponseAbstract

Description

```
<complexType name="ResponseAbstractType" abstract="true">
  <sequence>
     <element ref="ds:Signature" minOccurs="0"/>
  </sequence>
  <attribute name="ResponseID" type="saml:IDType" use="required"/>
  <attribute name="InResponseTo" type="saml:IDReferenceType" use="optional"/>
  <attribute name="MajorVersion" type="integer" use="required"/>
  <attribute name="MinorVersion" type="integer" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
  <attribute name="Recipient" type="anyURI" use="optional"/>
</complexType>

From oasis-sstc-saml-schema-assertion-1.0.xsd:
<simpleType name="IDType">
  <restriction base="string"/>
</simpleType>
<simpleType name="IDReferenceType">
  <restriction base="string"/>
</simpleType>
```

Figure 5.35: Schema fragment for samlp:ResponseAbstractType

Functions

lasso_samlp_response_abstract_fill ()

```
void
lasso_samlp_response_abstract_fill (LassoSamlpResponseAbstract *response,
                                    const char *InResponseTo,
                                    const char *Recipient);
```

Types and Values

struct LassoSamlpResponseAbstract

```
struct LassoSamlpResponseAbstract {
  LassoNode parent;

  /* <attribute name="ResponseID" type="saml:IDType" use="required"/> */
  char *ResponseID;
  /* <attribute name="InResponseTo" type="saml:IDReferenceType" use="optional"/> */
  char *InResponseTo;
  /* <attribute name="MajorVersion" type="integer" use="required"/> */
  int MajorVersion;
  /* <attribute name="MinorVersion" type="integer" use="required"/> */
  int MinorVersion;
  /* <attribute name="IssueInstant" type="dateTime" use="required"/> */
  char *IssueInstant;
  /* <attribute name="Recipient" type="anyURI" use="optional"/> */
  char *Recipient;

  /* ds:Signature stuffs */
  LassoSignatureType sign_type;
  LassoSignatureMethod sign_method;
  char *private_key_file;
  char *certificate_file;
};
```

5.40 LassoSamlpResponse

LassoSamlpResponse — <samlp:Response>

Functions

LassoNode *	lasso_samlp_response_new ()

Types and Values

struct	LassoSamlpResponse

Description

```
<element name="Response" type="samlp:ResponseType"/>
<complexType name="ResponseType">
  <complexContent>
    <extension base="samlp:ResponseAbstractType">
      <sequence>
        <element ref="samlp:Status"/>
        <element ref="saml:Assertion" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 5.36: Schema fragment for samlp:Response

Functions

lasso_samlp_response_new ()

```
LassoNode~*
lasso_samlp_response_new (void);
```

Creates a new LassoSamlpResponse object.

Returns

a newly created LassoSamlpResponse object

Types and Values

struct LassoSamlpResponse

```
struct LassoSamlpResponse {
  LassoSamlpResponseAbstract parent;

  /* <element ref="samlp:Status"/> */
  LassoSamlpStatus *Status;
  /* <element ref="saml:Assertion" minOccurs="0" maxOccurs="unbounded"/> */
  GList *Assertion; /* of LassoSamlAssertion */
};
```

5.41 LassoSamlpStatusCode

LassoSamlpStatusCode — <samlp:StatusCode>

Functions

LassoSamlpStatusCode *	lasso_samlp_status_code_new ()

Types and Values

struct	LassoSamlpStatusCode

Description

```

<element name="StatusCode" type="samlp:StatusCodeType"/>
<complexType name="StatusCodeType">
  <sequence>
    <element ref="samlp:StatusCode" minOccurs="0"/>
  </sequence>
  <attribute name="Value" type="QName" use="required"/>
</complexType>
```

Figure 5.37: Schema fragment for samlp:StatusCode

Functions

lasso_samlp_status_code_new ()

```
LassoSamlpStatusCode~*
lasso_samlp_status_code_new (void);
```

Creates a new LassoSamlpStatusCode object.

Returns

a newly created LassoSamlpStatusCode object

Types and Values

struct LassoSamlpStatusCode

```
struct LassoSamlpStatusCode {
  LassoNode parent;

  /* <element ref="samlp:StatusCode" minOccurs="0"/> */
  LassoSamlpStatusCode *StatusCode;
  /* <attribute name="Value" type="QName" use="required"/> */
  char *Value;
};
```

5.42 LassoSamlpStatus

LassoSamlpStatus — <samlp:Status>

Functions

LassoSamlpStatus *	lasso_samlp_status_new ()

Types and Values

struct	LassoSamlpStatus

Description

```
<element name="Status" type="samlp:StatusType"/>
<complexType name="StatusType">
  <sequence>
    <element ref="samlp:StatusCode"/>
    <element ref="samlp:StatusMessage" minOccurs="0" maxOccurs="1"/>
    <element ref="samlp:StatusDetail" minOccurs="0"/>
  </sequence>
</complexType>

<element name="StatusMessage" type="string"/>
```

Figure 5.38: Schema fragment for samlp:Status

Functions

lasso_samlp_status_new ()

```
LassoSamlpStatus~*
lasso_samlp_status_new (void);
```

Creates a new LassoSamlpStatus object.

Returns

a newly created LassoSamlpStatus object

Types and Values

struct LassoSamlpStatus

```
struct LassoSamlpStatus {
  LassoNode parent;

  /* <element ref="samlp:StatusCode"/> */
  LassoSamlpStatusCode *StatusCode;
  /* <element ref="samlp:StatusMessage" minOccurs="0" maxOccurs="1"/> */
  char *StatusMessage;
};
```

5.43 LassoSamlStatementAbstract

LassoSamlStatementAbstract — <saml:StatementAbstract>

Types and Values

struct	LassoSamlStatementAbstract

Description

```
<element name="Statement" type="saml:StatementAbstractType"/>
<complexType name="StatementAbstractType" abstract="true"/>
```

Figure 5.39: Schema fragment for saml:StatementAbstract

Functions

Types and Values

struct LassoSamlStatementAbstract

```
struct LassoSamlStatementAbstract {
  LassoNode parent;
};
```

5.44 LassoSamlSubjectConfirmation

LassoSamlSubjectConfirmation — <saml:SubjectConfirmation>

Functions

LassoSamlSubjectConfirmation *	lasso_saml_subject_confirmation_new ()

Types and Values

struct	LassoSamlSubjectConfirmation

Description

```

<element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
<complexType name="SubjectConfirmationType">
  <sequence>
    <element ref="saml:ConfirmationMethod" maxOccurs="unbounded"/>
    <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
    <element ref="ds:KeyInfo" minOccurs="0"/>
  </sequence>
</complexType>

<element name="SubjectConfirmationData" type="anyType"/>
<element name="ConfirmationMethod" type="anyURI"/>
```

Figure 5.40: Schema fragment for saml:SubjectConfirmation

Functions

lasso_saml_subject_confirmation_new ()

```
LassoSamlSubjectConfirmation~*
lasso_saml_subject_confirmation_new (void);
```

Creates a new LassoSamlSubjectConfirmation object.

Returns

a newly created LassoSamlSubjectConfirmation object

Types and Values

struct LassoSamlSubjectConfirmation

```
struct LassoSamlSubjectConfirmation {
  LassoNode parent;

  /* <element ref="saml:ConfirmationMethod" maxOccurs="unbounded"/> */
  GList *ConfirmationMethod; /* of strings */
  /* <element ref="saml:SubjectConfirmationData" minOccurs="0"/> */
  char *SubjectConfirmationData;
  LassoDsKeyInfo *KeyInfo;
};
```

5.45 LassoSamlSubjectLocality

LassoSamlSubjectLocality — <saml:SubjectLocality>

Functions

LassoNode *	lasso_saml_subject_locality_new ()

Types and Values

struct	LassoSamlSubjectLocality

Description

```
<element name="SubjectLocality" type="saml:SubjectLocalityType"/>
<complexType name="SubjectLocalityType">
  <attribute name="IPAddress" type="string" use="optional"/>
  <attribute name="DNSAddress" type="string" use="optional"/>
</complexType>
```

Figure 5.41: Schema fragment for saml:SubjectLocality

Functions

lasso_saml_subject_locality_new ()

```
LassoNode~*
lasso_saml_subject_locality_new (void);
```

Creates a new LassoSamlSubjectLocality object.

Returns

a newly created LassoSamlSubjectLocality object

Types and Values

struct LassoSamlSubjectLocality

```
struct LassoSamlSubjectLocality {
  LassoNode parent;

  /* <attribute name="IPAddress" type="string" use="optional"/> */
  char *IPAddress;
  /* <attribute name="DNSAddress" type="string" use="optional"/> */
  char *DNSAddress;
};
```

5.46 LassoSamlSubjectStatementAbstract

LassoSamlSubjectStatementAbstract — <saml:SubjectStatementAbstractType>

Types and Values

struct	LassoSamlSubjectStatementAbstract

Description

```
<complexType name="SubjectStatementAbstractType" abstract="true">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <sequence>
        <element ref="saml:Subject"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 5.42: Schema fragment for saml:SubjectStatementAbstractType

Functions

Types and Values

struct LassoSamlSubjectStatementAbstract

```
struct LassoSamlSubjectStatementAbstract {
  LassoSamlStatementAbstract parent;

  /* <element ref="saml:Subject"/> */
  LassoSamlSubject *Subject;
};
```

5.47 LassoSamlSubjectStatement

LassoSamlSubjectStatement —

Functions

LassoNode *	lasso_saml_subject_statement_new ()

Types and Values

struct	LassoSamlSubjectStatement

Description

Functions

lasso_saml_subject_statement_new ()

```
LassoNode~*
lasso_saml_subject_statement_new (void);
```

Creates a new LassoSamlSubjectStatement object.

Returns

a newly created LassoSamlSubjectStatement object

Types and Values

struct LassoSamlSubjectStatement

```
struct LassoSamlSubjectStatement {
  LassoSamlSubjectStatementAbstract parent;
};
```

5.48 LassoSamlSubject

LassoSamlSubject — <saml:Subject>

Functions

LassoNode *	lasso_saml_subject_new ()

Types and Values

struct	LassoSamlSubject

Description

```

<element name="Subject" type="saml:SubjectType"/>
<complexType name="SubjectType">
  <choice>
    <sequence>
      <element ref="saml:NameIdentifier"/>
      <element ref="saml:SubjectConfirmation" minOccurs="0"/>
    </sequence>
    <element ref="saml:SubjectConfirmation"/>
  </choice>
</complexType>
```

Figure 5.43: Schema fragment for saml:Subject

Functions

lasso_saml_subject_new ()

```
LassoNode~*
lasso_saml_subject_new (void);
```

Creates a new LassoSamlSubject object.

Returns

a newly created LassoSamlSubject object

Types and Values

struct LassoSamlSubject

```
struct LassoSamlSubject {
  LassoNode parent;

  LassoSamlNameIdentifier *NameIdentifier;
  LassoSamlSubjectConfirmation *SubjectConfirmation;
  LassoSaml2EncryptedElement *EncryptedNameIdentifier;
};
```

Chapter 6

SAML 2.0 Single Sign On profiles

The profile LassoLogin and LassoLogout are shared between SAML 2.0 and ID-FF 1.2, depending on the declared protocol support, Lasso will create request respecting the chosen standard. Beware that initialization of the LassoLogin object, after construction, differ between the two stacks of profiles. The LassoNameIdManagement profile replace the nearly equivalent LassoDefederation profile from ID-FF 1.2.

6.1 LassoAssertionQuery

LassoAssertionQuery —

Functions

LassoAssertionQuery *	lasso_assertion_query_new ()
void	lasso_assertion_query_destroy ()
lasso_error_t	lasso_assertion_query_init_request ()
lasso_error_t	lasso_assertion_query_validate_request ()
lasso_error_t	lasso_assertion_query_build_request_msg ()
lasso_error_t	lasso_assertion_query_process_request_msg ()
lasso_error_t	lasso_assertion_query_build_response_msg ()
lasso_error_t	lasso_assertion_query_process_response_msg ()
lasso_error_t	lasso_assertion_query_add_attribute_request ()
LassoAssertionQueryRequestType	lasso_assertion_query_get_request_type ()

Types and Values

enum	LassoAssertionQueryRequestType
struct	LassoAssertionQuery

Description

Functions

lasso_assertion_query_new ()

```
LassoAssertionQuery~*
lasso_assertion_query_new (LassoServer *server);
```

Creates a new LassoAssertionQuery.

Parameters

server	the LassoServer

Returns

a newly created LassoAssertionQuery object; or NULL if an error occured

lasso_assertion_query_destroy ()

```
void
lasso_assertion_query_destroy (LassoAssertionQuery *assertion_query);
```

Destroys a LassoAssertionQuery object.

Parameters

assertion_query	a LassoAssertionQuery

lasso_assertion_query_init_request ()

```
lasso_error_t
lasso_assertion_query_init_request (LassoAssertionQuery *assertion_query,
                                    char *remote_provider_id,
                                    LassoHttpMethod http_method,
                                    LassoAssertionQueryRequestType query_request_type);
```

Initializes a new Assertion Query Request. For the AssertionID request type, the remote_provider_id is mandatory, for all other kind of request it is optional if we can find a provider supporting the associated role, i.e. IDP; authentication, attribute and authorization authority.

Parameters

assertion_query	a LassoAssertionQuery	
remote_provider_id	the providerID of the remote provider.	*[allow-none]*
http_method	if set, then it get the protocol profile in metadata corresponding of this HTTP request method.	
query_request_type	the type of request.	

Returns

0 on success; or a negative value otherwise.

lasso_assertion_query_validate_request ()

```
lasso_error_t
```

```
lasso_assertion_query_validate_request
                               (LassoAssertionQuery *assertion_query);
```

Processes a Assertion query or request; caller must add assertions to the response afterwards.

Parameters

assertion_query	a LassoAssertionQuery	

Returns

0 on success; or a negative value otherwise.

lasso_assertion_query_build_request_msg ()

```
lasso_error_t
lasso_assertion_query_build_request_msg
                               (LassoAssertionQuery *assertion_query);
```

Build an Assertion Query profile request message.

Parameters

assertion_query	a LassoAssertionQuery	

Returns

0 on success; or a negative value otherwise.

lasso_assertion_query_process_request_msg ()

```
lasso_error_t
lasso_assertion_query_process_request_msg
                               (LassoAssertionQuery *assertion_query,
                                gchar *request_msg);
```

Processes a Assertion query or request message. Rebuilds a request object from the message and check its signature.

Parameters

assertion_query	a LassoAssertionQuery	
request_msg	the Assertion query or request message	

Returns

0 on success; or a negative value otherwise.

lasso_assertion_query_build_response_msg ()

```
lasso_error_t
lasso_assertion_query_build_response_msg
                               (LassoAssertionQuery *assertion_query);
```

Builds the Response message.

Parameters

assertion_query	a LassoAssertionQuery	

Returns

0 on success; or a negative value otherwise.

lasso_assertion_query_process_response_msg ()

```
lasso_error_t
lasso_assertion_query_process_response_msg
                               (LassoAssertionQuery *assertion_query,
                                gchar *response_msg);
```

Parses the response message and builds the corresponding response object.

Parameters

assertion_query	a LassoAssertionQuery	
response_msg	the response message	

Returns

0 on success; or a negative value otherwise.

lasso_assertion_query_add_attribute_request ()

```
lasso_error_t
lasso_assertion_query_add_attribute_request
                               (LassoAssertionQuery *assertion_query,
                                char *format,
                                char *name);
```

Append a new attribute designator to the current attribute request.

Parameters

assertion_query	a LassoAssertionQuery object	
attribute_format	the attribute designator format	
attribute_name	the attribute designator name	

Returns

0 if successful, an error code otherwise.

lasso_assertion_query_get_request_type ()

```
LassoAssertionQueryRequestType
lasso_assertion_query_get_request_type
                                (LassoAssertionQuery *assertion_query);
```

Return the type of the last processed request.

Parameters

assertion_query	a LassoAssertionQuery object

Returns

a LassoAssertionQueryRequestType value

Types and Values

enum LassoAssertionQueryRequestType

Enumerate the existing kind of AssertionQuery requests.

Members

LASSO_ASSERTION_QUERY_REQUEST_TYPE_UNSET	the unknown value
LASSO_ASSERTION_QUERY_REQUEST_TYPE_ASSERTION_ID	an AssertionID request, to retrieve an assertion by its ID.

LASSO_ASSERTION_QUERY_REQUEST_TYPE_AUTHN	an AuthnQuery request, which is used to request existing authentication assertions about a given subject from an Authentication Authority
LASSO_ASSERTION_QUERY_REQUEST_TYPE_ATTRIBUTE	an AttributeQuery, which is used to retrieve attribute an a principal.

LASSO_ASSERTION_QUERY_REQUEST_TYPE_AUTHZ_DECISION	an AuthzDecisionQuery, which is used to request authorisation to let a principal access a certain resource.
LASSO_ASSERTION_QUERY_REQUEST_TYPE_LAST	

struct LassoAssertionQuery

```
struct LassoAssertionQuery {
  LassoProfile parent;
};
```

6.2 LassoNameIdManagement

LassoNameIdManagement — Name Id Management Profile (SAMLv2)

Functions

LassoNameIdManagement *	lasso_name_id_management_new ()
LassoNameIdManagement *	lasso_name_id_management_new_from_dump ()
char *	lasso_name_id_management_dump ()
void	lasso_name_id_management_destroy ()
lasso_error_t	lasso_name_id_management_init_request ()
lasso_error_t	lasso_name_id_management_build_request_msg ()
lasso_error_t	lasso_name_id_management_process_request_msg ()
lasso_error_t	lasso_name_id_management_validate_request ()
lasso_error_t	lasso_name_id_management_build_response_msg ()
lasso_error_t	lasso_name_id_management_process_response_msg ()

Types and Values

struct	LassoNameIdManagement

Description

Functions

lasso_name_id_management_new ()

```
LassoNameIdManagement~*
lasso_name_id_management_new (LassoServer *server);
```

Creates a new LassoNameIdManagement.

Parameters

server	the LassoServer	

Returns

a newly created LassoNameIdManagement object; or NULL if an error occured

lasso_name_id_management_new_from_dump ()

```
LassoNameIdManagement~*
lasso_name_id_management_new_from_dump
                              (LassoServer *server,
                               const char *dump);
```

Restores the *dump* to a new LassoLogout.

Parameters

server	the LassoServer	
dump	XML name_id_management dump	

Returns

a newly created LassoLogout; or NULL if an error occured

lasso_name_id_management_dump ()

```
char~*
lasso_name_id_management_dump (LassoNameIdManagement *name_id_management);
```

Dumps *name_id_management* content to an XML string.

Parameters

name_id_management	a LassoLogout

Returns

the dump string. It must be freed by the caller.

[transfer full]

lasso_name_id_management_destroy ()

```
void
lasso_name_id_management_destroy (LassoNameIdManagement *name_id_management);
```

Destroys a LassoNameIdManagement object.

Parameters

name_id_management	a LassoNameIdManagement

lasso_name_id_management_init_request ()

```
lasso_error_t
lasso_name_id_management_init_request (LassoNameIdManagement *name_id_management,
                                       char *remote_provider_id,
                                       char *new_name_id,
                                       LassoHttpMethod http_method);
```

Initializes a new Name Id Management Request. If `new_name_id` is NULL, it is a Termination request, if not and we are an IdP is a NameID change request, if we are a SP, it is a request to add a SP provided Id to the NameID of the IdP. It can be useful if the SP do not want to store the federation, instead he can export its own identifiers to the IdP.

Parameters

name_id_management	a LassoNameIdManagement
remote_provider_id	the providerID of the remote provider.
new_name_id	the new NameId or NULL to terminate a federation
http_method	if set, then it get the protocol profile in metadata corresponding of this HTTP request method.

Returns

0 on success; or a negative value otherwise.

lasso_name_id_management_build_request_msg ()

```
lasso_error_t
lasso_name_id_management_build_request_msg
```

```
                                (LassoNameIdManagement *name_id_management);
```

Builds the Name Id Management request message.

Parameters

name_id_management	a LassoNameIdManagement

Returns

0 on success; or a negative value otherwise.

lasso_name_id_management_process_request_msg ()

```
lasso_error_t
lasso_name_id_management_process_request_msg
                                (LassoNameIdManagement *name_id_management,
                                 gchar *request_msg);
```

Processes a Name Id Management request message. Rebuilds a request object from the message and check its signature.

Parameters

name_id_management	a LassoNameIdManagement
request_msg	the Name Id Management request message

Returns

0 on success; or a negative value otherwise.

lasso_name_id_management_validate_request ()

```
lasso_error_t
lasso_name_id_management_validate_request
                                (LassoNameIdManagement *name_id_management);
```

Processes a Name Id Management request, performing requested actions against principal federations. Profile identity may have to be saved afterwards.

Parameters

name_id_management	a LassoNameIdManagement

Returns

0 on success; or a negative value otherwise.

lasso_name_id_management_build_response_msg ()

```
lasso_error_t
lasso_name_id_management_build_response_msg
                               (LassoNameIdManagement *name_id_management);
```

Builds the Name Id Management response message.

Parameters

name_id_management	a LassoNameIdManagement

Returns

0 on success; or a negative value otherwise.

lasso_name_id_management_process_response_msg ()

```
lasso_error_t
lasso_name_id_management_process_response_msg
                               (LassoNameIdManagement *name_id_management,
                                gchar *response_msg);
```

Parses the response message and builds the corresponding response object. Performs requested actions against principal federations. Profile identity may have to be saved afterwards.

Parameters

name_id_management	a LassoNameIdManagement
response_msg	the response message

Returns

0 on success; or a negative value otherwise.

Types and Values

struct LassoNameIdManagement

```
struct LassoNameIdManagement {
  LassoProfile parent;
};
```

6.3 LassoEcp

LassoEcp — Enhanced Client or Proxy Profile (SAMLv2)

Functions

gboolean	lasso_ecp_is_provider_in_sp_idplist ()
gboolean	lasso_ecp_is_idp_entry_known_idp_supporting_ecp ()
void	lasso_ecp_set_known_sp_provided_idp_entries_supporting_ecp ()
gboolean	lasso_ecp_has_sp_idplist ()
gchar *	lasso_ecp_get_endpoint_url_by_entity_id ()
int	lasso_ecp_process_sp_idp_list ()
LassoEcp *	lasso_ecp_new ()
lasso_error_t	lasso_ecp_process_authn_request_msg ()
lasso_error_t	lasso_ecp_process_response_msg ()
void	lasso_ecp_destroy ()

Types and Values

struct	LassoEcp

Description

Introduction

The LassoEcp object is used to implement a SAMLv2 ECP client. If you want to support ECP in a SP see [ecp-sp]. If you want to support ECP in a IdP see [ecp-idp].

ECP Operational Steps

SAML2 Profile for ECP (Section 4.2) defines these steps for an ECP transaction

1. ECP issues HTTP Request to SP
2. SP issues <samlp:AuthnRequest> to ECP using PAOS
3. ECP determines IdP
4. ECP conveys <samlp:AuthnRequest> to IdP using SOAP
5. IdP identifies principal
6. IdP issues <samlp:Response> to ECP, targeted at SP using SOAP
7. ECP conveys <samlp:Response> to SP using PAOS
8. SP grants or denies access to principal

Functions

lasso_ecp_is_provider_in_sp_idplist ()

```
gboolean
lasso_ecp_is_provider_in_sp_idplist (LassoEcp *ecp,
                                     const gchar *entity_id);
```

Check to see if the provider with `entity_id` is in the ecp IDPList returned by the SP.

Parameters

ecp	a LassoEcp	
entity_id	EntityID to check if member of LassoEcp.IDPList	

Returns

TRUE if *entity_id* is in LassoEcp.IDPList, FALSE otherwise

lasso_ecp_is_idp_entry_known_idp_supporting_ecp ()

```
gboolean
lasso_ecp_is_idp_entry_known_idp_supporting_ecp
                                (LassoEcp *ecp,
                                 const LassoSamlp2IDPEntry *idp_entry);
```

Check to see if the *idp_entry* is in the *entity_id_list*

Parameters

ecp	a LassoEcp	
idp_entry	LassoSamlp2IDPEntry to check if member of *entity_id_list*	

Returns

TRUE if *entity_id* is in *idp_list* , FALSE otherwise

lasso_ecp_set_known_sp_provided_idp_entries_supporting_ecp ()

```
void
lasso_ecp_set_known_sp_provided_idp_entries_supporting_ecp
                                (LassoEcp *ecp);
```

The SP may provide a list of LassoSamlp2IDPEntry (LassoEcp.sp_idp_list) which it trusts. The ECP client has a list of IDP EntityID's it knows supports ECP (LassoEcp.known_idp_entity_ids_supporting_ecp). The set of possible IDP's which can service the SP's authn request are the interesection of these two lists (the IDP's the SP approves and IDP's the ECP knows about). This find the common members between the two lists and assign them to LassoEcp.known_sp_provided_idp_entries_supporting_ecp.

Parameters

ecp	a LassoEcp	

lasso_ecp_has_sp_idplist ()

```
gboolean
lasso_ecp_has_sp_idplist (LassoEcp *ecp);
```

Returns TRUE if the SP provided an IDP List, FALSE otherwise.

Parameters

ecp	a LassoEcp

lasso_ecp_get_endpoint_url_by_entity_id ()

```
gchar~*
lasso_ecp_get_endpoint_url_by_entity_id
                                (LassoEcp *ecp,
                                 const gchar *entity_id);
```

Returns the SingleSignOnService SOAP endpoint URL for the specified *entity_id* . If the provider cannot be found or if the provider does not have a matching endpoint NULL will be returned.

Parameters

ecp	a LassoEcp
entity_id	the EntityID of the IdP

Returns

url (must be freed by caller)

lasso_ecp_process_sp_idp_list ()

```
int
lasso_ecp_process_sp_idp_list (LassoEcp *ecp,
                               const LassoSamlp2IDPList *sp_idp_list);
```

The SP may optionally send a list of IdP's it trusts in ecp:IDPList. The ecp:IDPList may not be complete if the IDPList.GetComplete is non-NULL. If so the IDPList.GetComplete is a URL where a complete IDPList may be fetched.

Whenever the IDPList is updated this function needs to be called because it sets the LassoEcp.known_sp_provided_idp_entries_supportin and the default IdP URL (LassoProfile.msg_url).

The LassoEcp client has a list of IdP's it knows supports ECP (LassoEcp.known_idp_entity_ids_supporting_ecp). The set of IdP's available to select from should be those in common between SP provided IdP list and those known by this ECP client to support ECP.

This routine sets the LassoEcp.known_sp_provided_idp_entries_supporting_ecp list to the common members (e.g. intersection) of the SP provided IdP list and the list of known IdP's supporting ECP.

A default IdP will be selected and it's endpoint URL will be assigned to LassoProfile.msg_url.

If the SP provided an IDP list then the default URL will be taken from first IDPEntry in LassoEcp.known_sp_provided_idp_entries_suppo otherwise it will be taken from LassoEcp.known_idp_entity_ids_supporting_ecp.

Parameters

ecp	a LassoEcp

lasso_ecp_new ()

```
LassoEcp~*
lasso_ecp_new (LassoServer *server);
```

Creates a new LassoEcp.

Returns

a newly created LassoEcp object; or NULL if an error occured

lasso_ecp_process_authn_request_msg ()

```
lasso_error_t
lasso_ecp_process_authn_request_msg (LassoEcp *ecp,
                                     const char *authn_request_msg);
```

This function implements the following ECP step: ECP Step 3, ECP determines IdP ECP Step 4, parse SP PAOS Authn request, build SOAP for IdP

This is to be used in an ECP client. The *authn_request_msg* is the SOAP PAOS message received from the SP in response to a resource request with an HTTP Accept header indicating PAOS support.

The following actions are implemented:

- Extract the samlp:AuthnRequest from the SOAP body and build a new SOAP message containing the samlp:AuthnRequest which will be forwarded to the IdP. This new SOAP message is stored in the LassoProfile.msg_body.
- Parse the SOAP header which will contain a paos:Request, a ecp:Request and optionally a ecp:RelayState. Some of the data in these headers need to be preserved for later processing steps.
  1. The paos:Request.responseConsumerURL is copied to the LassoEcp.response_consumer_url. This is necessary because the ECP client MUST assure it matches the ecp:Response.AssertionConsumerServiceURL returned by the IdP to prevent man-in-the-middle attacks. It must also match the samlp:AuthnRequest.AssertionConsumerServiceURL.
  2. If the paos:Request contained a messageID it is copied to LassoEcp.message_id so it can be returned in the subsequent paos:Response.refToMessageID. This allows a provider to correlate messages.
  3. If an ecp:RelayState is present it is copied to LassoEcp.relaystate. This is necessary because in step 7 when the ECP responds to the SP it must include RelayState provided in the request.
- In addition the following items are copied to the LassoEcp for informational purposes:
  - LassoEcp.issuer = ecp:Request.Issuer
  - LassoEcp.provider_name = ecp:Request.ProviderName
  - LassoEcp.is_passive = ecp:Request.IsPassive
  - LassoEcp.sp_idp_list = ecp:Request.IDPList

IdP Selection

In Step 3. The ECP must determine the IdP to forward the AuthnRequest to. There are two sets of IdP's which come into play. The ECP client has a set of IdP's it knows about because their metadata has been loaded into the LassoServer object. The SP may optionally send a list of IdP's in the ecp:Request that it trusts.

The selected IdP *must* be one of the IdP's loaded into the LassoServer object from metadata because the IdP endpoints must be known. Furthermore the IdP *must* support the SingleSignOnService using the SOAP binding. Therefore the known IdP's are filtered for those that match this criteria and a list of their EntityID's are assigned to LassoEcp.known_idp_entity_ids_supporting_ecp. The selected IdP *must* be a member of this list.

The SP may optionally send a list of IdP's it trusts. If the SP sends an IDPList the selected IdP should be a member of this list and from above we know it must also be a member of the LassoEcp.known_idp_entity_ids_supporting_ecp. Therefore the LassoEcp.known_sp_provided_idp_entries_supporting_ecp list is set to the common members (e.g. intersection) of the SP provided IdP list and the list of known IdP's supporting ECP.

When making an IdP selection if the SP provided an IdP List (use LassoEcp.lasso_ecp_has_sp_idplist()) then it should be selected from the LassoEcp.known_sp_provided_idp_entries_supporting_ecp list. Otherwise the IdP should be selected from LassoEcp.known_idp_entity_ids_supporting_ecp.

A default IdP will be selected using the above logic by picking the first IdP in the appropriate list, it's endpoint URL will be assigned to LassoProfile.msg_url. The above processing is implemented by LassoEcp.lasso_ecp_process_sp_idp_list() and if the SP IDPList is updated this routine should be called.

A note about the 3 IdP lists. The LassoEcp.sp_idp_list.IDPList and LassoEcp.known_sp_provided_idp_entries_supporting_ecp are GList's of LassoSamlp2IDPEntry object which have a ProviderID, Name, and Loc attribute. You may wish to use this SP provided information when making a decision or presenting in a user interface that allows a user to make a choice. The LassoEcp.known_idp_entity_ids_supporting_ecp is a GList of EntityID strings.

Given the EntityID of an IdP you can get the ECP endpoint by calling LassoEcp.lasso_ecp_get_endpoint_url_by_entity_id()

Results

After a successful return from this call you are ready to complete Step 4. and forward the request the IdP.

The URL to send to the request to will be LassoProfile.msg_url (if you accept the default IdP) and the body of the message to post will be LassoProfile.msg_body.

Side Effects

After a successful return the LassoEcp object will be updated with:

- ecp->response_consumer_url = paos_request->responseConsumerURL
- ecp->message_id = paos_request->messageID
- ecp->relaystate = ecp_relaystate->RelayState
- ecp->issuer = ecp_request->Issue
- ecp->provider_name = ecp_request->ProviderName
- ecp->is_passive = ecp_request->IsPassive
- ecp->known_idp_entity_ids_supporting_ecp
- ecp->sp_idp_list = ecp_request->IDPList
- ecp->known_sp_provided_idp_entries_supporting_ecp

Parameters

ecp	this LassoEcp object	
authn_request_msg	the PAOS authn request received from the SP	

lasso_ecp_process_response_msg ()

```
lasso_error_t
lasso_ecp_process_response_msg (LassoEcp *ecp,
                                const char *response_msg);
```

The function implements ECP Step 7; parse IdP SOAP response and build PAOS response for SP.

See SAML Profile Section 4.2.4.5 PAOS Response Header Block: ECP to SP

This is to be used in an ECP client. The *response_msg* parameter contains the SOAP response from the IdP. We extract the ECP Header Block and body from it. We will generate a new PAOS message to send to the SP, the SOAP header will contain a paos:Response. If we received a paos:Request.MessageID in Step. 4 from the SP then we will copy it back to the paos:Response.refToMessageID. If we received a RelayState we will add that to the SOAP header as well.

To prevent a man-in-the-middle attack we verify the responseConsumerURL we received in Step 4 matches the ecp:Response.AssertionC we just received back from the IdP. If they do not match we return a LASSO_ECP_ERROR_ASSERTION_CONSUMER_URL_MISMA error and set the LassoProvider.msg_body to the appropriate SOAP fault.

The new PAOS message for the SP we are buiding contains the IdP response in the new SOAP body and the new SOAP headers will contain a paso:Response and optionally an ecp:RelayState.

After a successful return from this call you are ready to complete Step 7. and forward the response to the SP.

The PASO message is assigned to the LassoProvider.msg_body and the desination URL is assigned to the LassoProvider.msg_url.

Side Effects

After a successful return the LassoEcp object will be updated with:

- ecp->assertion_consumer_url = ecp_response->AssertionConsumerServiceURL
- ecp.profile.msg_url = ecp->assertion_consumer_url
- ecp.profile.msg_body_url = PAOS response to SP

Parameters

ecp	this LassoEcp object	
response_msg	the SOAP response from the IdP	

lasso_ecp_destroy ()

```
void
lasso_ecp_destroy (LassoEcp *ecp);
```

Destroys a LassoEcp object

Parameters

ecp	a LassoEcp	

Types and Values

struct LassoEcp

```
struct LassoEcp {
  LassoProfile parent;

  gchar *assertion_consumer_url;
  gchar *message_id;
  gchar *response_consumer_url;
  gchar *relaystate;
  LassoSaml2NameID *issuer;
  gchar *provider_name;
  gboolean is_passive;
  LassoSamlp2IDPList *sp_idp_list;
  GList *known_sp_provided_idp_entries_supporting_ecp; /* of LassoSamlp2IDPEntry */
  GList *known_idp_entity_ids_supporting_ecp;          /* of strings */
};
```

6.4 Utility functions for SAML 2.0

Utility functions for SAML 2.0 — Misc functions used inside Lasso

Stability Level

Internal, unless otherwise indicated

Functions

char *	lasso_build_unique_id ()
gboolean	lasso_profile_is_saml_query ()
LassoSaml2EncryptedElement *	lasso_provider_saml2_node_encrypt ()

Description

Functions

lasso_build_unique_id ()

```
char~*
lasso_build_unique_id (unsigned int size);
```

Builds an ID which has an unicity probability of 2^(-size*4).

Parameters

size	the ID’s length (between 32 and 40)

Returns

a "unique" ID (begin always with _ character).

[transfer full]

lasso_profile_is_saml_query ()

```
gboolean
lasso_profile_is_saml_query (const gchar *query);
```

lasso_provider_saml2_node_encrypt ()

```
LassoSaml2EncryptedElement~*
lasso_provider_saml2_node_encrypt (const LassoProvider *provider,
                                   LassoNode *lasso_node);
```

Dump the node object to an XML fragment, then encrypt this fragment using encryption key of *provider* , then encapsulate the resulting encrypted content into a LassoSaml2EncryptedElement.

Parameters

provider	a LassoProvider object	
lasso_node	a LassoNode object	

Returns

a newly created LassoSaml2EncryptedElement if successfull, NULL otherwise.

Chapter 7

Objects from SAML 2.0 schemas

7.1 SAML 2.0 Strings

SAML 2.0 Strings — String constants from SAML 2.0 specifications

Stability Level

Stable, unless otherwise indicated

Types and Values

#define	LASSO_SAML2_METADATA_HREF
#define	LASSO_SAML2_METADATA_PREFIX
#define	LASSO_SAML2_PROTOCOL_HREF
#define	LASSO_SAML2_PROTOCOL_PREFIX
#define	LASSO_SAML2_ASSERTION_HREF
#define	LASSO_SAML2_ASSERTION_PREFIX
#define	LASSO_SAML2_METADATA_BINDING_SOAP
#define	LASSO_SAML2_METADATA_BINDING_REDIRECT
#define	LASSO_SAML2_METADATA_BINDING_POST
#define	LASSO_SAML2_METADATA_BINDING_ARTIFACT
#define	LASSO_SAML2_METADATA_BINDING_PAOS
#define	LASSO_SAML2_DEFLATE_ENCODING
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_UNSPECIFIED
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_EMAIL
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_X509
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_WINDOWS
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_KERBEROS
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENTITY
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_PERSISTENT
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_TRANSIENT
#define	LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENCRYPTED
#define	LASSO_SAML2_ATTRIBUTE_NAME_EPR
#define	LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_UNSPECIFIED
#define	LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_URI
#define	LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_BASIC
#define	LASSO_SAML2_ACTION_NAMESPACE_RWEDC
#define	LASSO_SAML2_ACTION_NAMESPACE_RWEDC_NEGATION

#define	LASSO_SAML2_ACTION_NAMESPACE_GHPP
#define	LASSO_SAML2_ACTION_NAMESPACE_UNIX
#define	LASSO_SAML2_ACTION_RWEDC_READ
#define	LASSO_SAML2_ACTION_RWEDC_WRITE
#define	LASSO_SAML2_ACTION_RWEDC_EXECUTE
#define	LASSO_SAML2_ACTION_RWEDC_DELETE
#define	LASSO_SAML2_ACTION_RWEDC_CONTROL
#define	LASSO_SAML2_ACTION_RWEDC_NEGATION
#define	LASSO_SAML2_ACTION_GHPP_GET
#define	LASSO_SAML2_ACTION_GHPP_HEAD
#define	LASSO_SAML2_ACTION_GHPP_PUT
#define	LASSO_SAML2_ACTION_GHPP_POST
#define	LASSO_SAML2_CONSENT_OBTAINED
#define	LASSO_SAML2_CONSENT_PRIOR
#define	LASSO_SAML2_CONSENT_IMPLICIT
#define	LASSO_SAML2_CONSENT_EXPLICIT
#define	LASSO_SAML2_CONSENT_UNAVAILABLE
#define	LASSO_SAML2_CONSENT_INAPPLICABLE
#define	LASSO_SAML2_STATUS_CODE_SUCCESS
#define	LASSO_SAML2_STATUS_CODE_REQUESTER
#define	LASSO_SAML2_STATUS_CODE_RESPONDER
#define	LASSO_SAML2_STATUS_CODE_VERSION_MISMATCH
#define	LASSO_SAML2_STATUS_CODE_AUTHN_FAILED
#define	LASSO_SAML2_STATUS_CODE_INVALID_ATTR_NAME
#define	LASSO_SAML2_STATUS_CODE_INVALID_NAME_ID_POLICY
#define	LASSO_SAML2_STATUS_CODE_NO_AUTHN_CONTEXT
#define	LASSO_SAML2_STATUS_CODE_NO_AVAILABLE_IDP
#define	LASSO_SAML2_STATUS_CODE_NO_PASSIVE
#define	LASSO_SAML2_STATUS_CODE_NO_SUPPORTED_IDP
#define	LASSO_SAML2_STATUS_CODE_PARTIAL_LOGOUT
#define	LASSO_SAML2_STATUS_CODE_PROXY_COUNT_EXCEEDED
#define	LASSO_SAML2_STATUS_CODE_REQUEST_DENIED
#define	LASSO_SAML2_STATUS_CODE_REQUEST_UNSUPPORTED
#define	LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_DEPRECATED
#define	LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_TOO_HIGH
#define	LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_TOO_LOW
#define	LASSO_SAML2_STATUS_CODE_RESOURCE_NOT_RECOGNIZED
#define	LASSO_SAML2_STATUS_CODE_TOO_MANY_RESPONSES
#define	LASSO_SAML2_STATUS_CODE_UNKNOWN_ATTR_PROFILE
#define	LASSO_SAML2_STATUS_CODE_UNKNOWN_PRINCIPAL
#define	LASSO_SAML2_STATUS_CODE_UNSUPPORTED_BINDING
#define	LASSO_SAML2_AUTHN_CONTEXT_AUTHENTICATED_TELEPHONY
#define	LASSO_SAML2_AUTHN_CONTEXT_INTERNET_PROTOCOL
#define	LASSO_SAML2_AUTHN_CONTEXT_INTERNET_PROTOCOL_PASSWORD
#define	LASSO_SAML2_AUTHN_CONTEXT_KERBEROS
#define	LASSO_SAML2_AUTHN_CONTEXT_MOBILE_ONE_FACTOR_CONTRACT
#define	LASSO_SAML2_AUTHN_CONTEXT_MOBILE_ONE_FACTOR_UNREGISTERED
#define	LASSO_SAML2_AUTHN_CONTEXT_MOBILE_TWO_FACTOR_CONTRACT
#define	LASSO_SAML2_AUTHN_CONTEXT_MOBILE_TWO_FACTOR_UNREGISTERED
#define	LASSO_SAML2_AUTHN_CONTEXT_NOMAD_TELEPHONY
#define	LASSO_SAML2_AUTHN_CONTEXT_PERSONALIZED_TELEPHONY
#define	LASSO_SAML2_AUTHN_CONTEXT_PGP
#define	LASSO_SAML2_AUTHN_CONTEXT_PASSWORD_PROTECTED_TRANSPORT
#define	LASSO_SAML2_AUTHN_CONTEXT_PASSWORD
#define	LASSO_SAML2_AUTHN_CONTEXT_PREVIOUS_SESSION
#define	LASSO_SAML2_AUTHN_CONTEXT_SMARTCARD
#define	LASSO_SAML2_AUTHN_CONTEXT_SMARTCARD_PKI

#define	LASSO_SAML2_AUTHN_CONTEXT_SOFTWARE_PKI
#define	LASSO_SAML2_AUTHN_CONTEXT_SPKI
#define	LASSO_SAML2_AUTHN_CONTEXT_SECURE_REMOTE_PASSWORD
#define	LASSO_SAML2_AUTHN_CONTEXT_TLS_CLIENT
#define	LASSO_SAML2_AUTHN_CONTEXT_X509
#define	LASSO_SAML2_AUTHN_CONTEXT_TELEPHONY
#define	LASSO_SAML2_AUTHN_CONTEXT_TIME_SYNC_TOKEN
#define	LASSO_SAML2_AUTHN_CONTEXT_XMLDSIG
#define	LASSO_SAML2_AUTHN_CONTEXT_UNSPECIFIED
#define	LASSO_SAML2_CONFIRMATION_METHOD_BEARER
#define	LASSO_SAML2_CONFIRMATION_METHOD_HOLDER_OF_KEY
#define	LASSO_SAML2_FIELD_RESPONSE
#define	LASSO_SAML2_FIELD_REQUEST
#define	LASSO_SAML2_FIELD_ARTIFACT
#define	LASSO_SAML2_FIELD_RELAYSTATE
#define	LASSO_SAML2_FIELD_SIGNATURE
#define	LASSO_SAML2_FIELD_SIGALG

Includes

```
#include <lasso/xml/saml-2.0/saml2_strings.h>
```

Description

Functions

Types and Values

LASSO_SAML2_METADATA_HREF

```
#define LASSO_SAML2_METADATA_HREF "urn:oasis:names:tc:SAML:2.0:metadata"
```

Namespace for SAML 2.0 metadata

LASSO_SAML2_METADATA_PREFIX

```
#define LASSO_SAML2_METADATA_PREFIX "md"
```

Preferred prefix for namespace of SAML 2.0 metadata

LASSO_SAML2_PROTOCOL_HREF

```
#define LASSO_SAML2_PROTOCOL_HREF "urn:oasis:names:tc:SAML:2.0:protocol"
```

Namespace for SAML 2.0 protocol.

LASSO_SAML2_PROTOCOL_PREFIX

```
#define LASSO_SAML2_PROTOCOL_PREFIX "samlp"
```

Preferred prefix for namespace of SAML 2.0 protocol

LASSO_SAML2_ASSERTION_HREF

```
#define LASSO_SAML2_ASSERTION_HREF "urn:oasis:names:tc:SAML:2.0:assertion"
```

Namespace for SAML 2.0 assertion

LASSO_SAML2_ASSERTION_PREFIX

```
#define LASSO_SAML2_ASSERTION_PREFIX "saml"
```

Preferred prefix for namespace of SAML 2.0 assertion

LASSO_SAML2_METADATA_BINDING_SOAP

```
#define LASSO_SAML2_METADATA_BINDING_SOAP "urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
```

URI for the SOAP binding.

LASSO_SAML2_METADATA_BINDING_REDIRECT

```
#define LASSO_SAML2_METADATA_BINDING_REDIRECT "urn:oasis:names:tc:SAML:2.0:bindings:HTTP- ←
    Redirect"
```

URI for the HTTP-Redirect binding.

LASSO_SAML2_METADATA_BINDING_POST

```
#define LASSO_SAML2_METADATA_BINDING_POST "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
```

URI for the HTTP-Post binding.

LASSO_SAML2_METADATA_BINDING_ARTIFACT

```
#define LASSO_SAML2_METADATA_BINDING_ARTIFACT "urn:oasis:names:tc:SAML:2.0:bindings:HTTP- ←
    Artifact"
```

URI for the HTTP-Artifact binding.

LASSO_SAML2_METADATA_BINDING_PAOS

```
#define LASSO_SAML2_METADATA_BINDING_PAOS "urn:oasis:names:tc:SAML:2.0:bindings:PAOS"
```

URI for the PAOS (or reverse SOAP) binding.

LASSO_SAML2_DEFLATE_ENCODING

```
#define LASSO_SAML2_DEFLATE_ENCODING "urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding: ←
    DEFLATE"
```

URI for URL-Encoding of kind DEFLATE (compress message content before encoding in the URI).

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_UNSPECIFIED

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_UNSPECIFIED
```

Name identifier format for local names, or free format name.

From saml-core-2.0-os.pdf:

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_EMAIL

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_EMAIL
```

Name identifier format for email addresses.

From saml-core-2.0-os.pdf:

> *domain*

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_X509

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_X509
```

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_WINDOWS

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_WINDOWS
```

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_KERBEROS

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_KERBEROS
```

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENTITY

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENTITY
```

Name identifier format for SAML 2.0 entities, i.e. identity and service providers.

From saml-core-2.0-os.pdf:

> Indicates that the content of the element is the identifier of an entity that provides SAML-based services (such as a SAML authority, requester, or responder) or is a participant in SAML profiles (such as a service provider supporting the browser SSO profile). Such an identifier can be used in the <Issuer> element to identify the issuer of a SAML request, response, or assertion, or within the <NameID> element to make assertions about system entities that can issue SAML requests, responses, and assertions. It can also be used in other elements and attributes whose purpose is to identify a system entity in various protocol exchanges.
>
> The syntax of such an identifier is a URI of not more than 1024 characters in length. It is RECOMMENDED that a system entity use a URL containing its own domain name to identify itself.
>
> The NameQualifier, SPNameQualifier, and SPProvidedID attributes MUST be omitted.

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_PERSISTENT

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_PERSISTENT
```

Name identifier format for SAML 2.0 federation.

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_TRANSIENT

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_TRANSIENT
```

Name identifier format for temporary SAML 2.0 federation.

LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENCRYPTED

```
#define             LASSO_SAML2_NAME_IDENTIFIER_FORMAT_ENCRYPTED
```

LASSO_SAML2_ATTRIBUTE_NAME_EPR

```
#define LASSO_SAML2_ATTRIBUTE_NAME_EPR "urn:liberty:disco:2006-08:DiscoveryEPR"
```

Attribute name for tranmitting Discovery bootstrap EPR when using ID-WSF 2.0 framework. It must be used conjointly with LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_URI as format for the attribute element.

LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_UNSPECIFIED

```
#define LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_UNSPECIFIED "urn:oasis:names:tc:SAML:2.0:attrname ↩
    -format:unspecified"
```

Attribute format whose interpretation is left to individual implementations.

LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_URI

```
#define LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_URI "urn:oasis:names:tc:SAML:2.0:attrname-format: ↩
    uri"
```

From saml-core-2.0-os.pdf:

LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_BASIC

```
#define LASSO_SAML2_ATTRIBUTE_NAME_FORMAT_BASIC "urn:oasis:names:tc:SAML:2.0:attrname- ↩
    format:basic"
```

Attribute format whose names are in the xs:Name domain.

LASSO_SAML2_ACTION_NAMESPACE_RWEDC

```
#define LASSO_SAML2_ACTION_NAMESPACE_RWEDC "urn:oasis:names:tc:SAML:1.0:action:rwedc"
```

Namespace for actions among: Read, Write, Execute, Delete and Control.

LASSO_SAML2_ACTION_NAMESPACE_RWEDC_NEGATION

```
#define LASSO_SAML2_ACTION_NAMESPACE_RWEDC_NEGATION "urn:oasis:names:tc:SAML:1.0:action: ↩
    rwedc-negation"
```

Namespace for actions among: Read, Write, Execute, Delete and Control and their negations, ~Read, ~Write, ~Execute, ~Delete, ~Control.

LASSO_SAML2_ACTION_NAMESPACE_GHPP

```
#define LASSO_SAML2_ACTION_NAMESPACE_GHPP "urn:oasis:names:tc:SAML:1.0:action:ghpp"
```

Namespace for actions among: GET, HEAD, PUT, POST.

LASSO_SAML2_ACTION_NAMESPACE_UNIX

```
#define LASSO_SAML2_ACTION_NAMESPACE_UNIX "urn:oasis:names:tc:SAML:1.0:action:unix"
```

Namespace for actions represented by a four digit numeric code in octal value, as Unix file permissions codes.

LASSO_SAML2_ACTION_RWEDC_READ

```
#define LASSO_SAML2_ACTION_RWEDC_READ "Read"
```

LASSO_SAML2_ACTION_RWEDC_WRITE

```
#define LASSO_SAML2_ACTION_RWEDC_WRITE "Write"
```

LASSO_SAML2_ACTION_RWEDC_EXECUTE

```
#define LASSO_SAML2_ACTION_RWEDC_EXECUTE "Execute"
```

LASSO_SAML2_ACTION_RWEDC_DELETE

```
#define LASSO_SAML2_ACTION_RWEDC_DELETE "Delete"
```

LASSO_SAML2_ACTION_RWEDC_CONTROL

```
#define LASSO_SAML2_ACTION_RWEDC_CONTROL "Control"
```

LASSO_SAML2_ACTION_RWEDC_NEGATION

```
#define LASSO_SAML2_ACTION_RWEDC_NEGATION "~"
```

LASSO_SAML2_ACTION_GHPP_GET

```
#define LASSO_SAML2_ACTION_GHPP_GET "GET"
```

LASSO_SAML2_ACTION_GHPP_HEAD

```
#define LASSO_SAML2_ACTION_GHPP_HEAD "HEAD"
```

LASSO_SAML2_ACTION_GHPP_PUT

```
#define LASSO_SAML2_ACTION_GHPP_PUT "PUT"
```

LASSO_SAML2_ACTION_GHPP_POST

```
#define LASSO_SAML2_ACTION_GHPP_POST "POST"
```

LASSO_SAML2_CONSENT_OBTAINED

```
#define LASSO_SAML2_CONSENT_OBTAINED "urn:oasis:names:tc:SAML:2.0:consent:obtained"
```

LASSO_SAML2_CONSENT_PRIOR

```
#define LASSO_SAML2_CONSENT_PRIOR "urn:oasis:names:tc:SAML:2.0:consent:prior"
```

LASSO_SAML2_CONSENT_IMPLICIT

```
#define LASSO_SAML2_CONSENT_IMPLICIT "urn:oasis:names:tc:SAML:2.0:consent:current-implicit"
```

LASSO_SAML2_CONSENT_EXPLICIT

```
#define LASSO_SAML2_CONSENT_EXPLICIT "urn:oasis:names:tc:SAML:2.0:consent:current-explicit"
```

LASSO_SAML2_CONSENT_UNAVAILABLE

```
#define LASSO_SAML2_CONSENT_UNAVAILABLE "urn:oasis:names:tc:SAML:2.0:consent:unavailable"
```

LASSO_SAML2_CONSENT_INAPPLICABLE

```
#define LASSO_SAML2_CONSENT_INAPPLICABLE "urn:oasis:names:tc:SAML:2.0:consent:inapplicable"
```

LASSO_SAML2_STATUS_CODE_SUCCESS

```
#define LASSO_SAML2_STATUS_CODE_SUCCESS "urn:oasis:names:tc:SAML:2.0:status:Success"
```

LASSO_SAML2_STATUS_CODE_REQUESTER

```
#define LASSO_SAML2_STATUS_CODE_REQUESTER "urn:oasis:names:tc:SAML:2.0:status:Requester"
```

LASSO_SAML2_STATUS_CODE_RESPONDER

```
#define LASSO_SAML2_STATUS_CODE_RESPONDER "urn:oasis:names:tc:SAML:2.0:status:Responder"
```

LASSO_SAML2_STATUS_CODE_VERSION_MISMATCH

```
#define             LASSO_SAML2_STATUS_CODE_VERSION_MISMATCH
```

LASSO_SAML2_STATUS_CODE_AUTHN_FAILED

```
#define LASSO_SAML2_STATUS_CODE_AUTHN_FAILED "urn:oasis:names:tc:SAML:2.0:status: ↩
    AuthnFailed"
```

LASSO_SAML2_STATUS_CODE_INVALID_ATTR_NAME

```
#define             LASSO_SAML2_STATUS_CODE_INVALID_ATTR_NAME
```

LASSO_SAML2_STATUS_CODE_INVALID_NAME_ID_POLICY

```
#define             LASSO_SAML2_STATUS_CODE_INVALID_NAME_ID_POLICY
```

LASSO_SAML2_STATUS_CODE_NO_AUTHN_CONTEXT

```
#define             LASSO_SAML2_STATUS_CODE_NO_AUTHN_CONTEXT
```

LASSO_SAML2_STATUS_CODE_NO_AVAILABLE_IDP

```
#define             LASSO_SAML2_STATUS_CODE_NO_AVAILABLE_IDP
```

LASSO_SAML2_STATUS_CODE_NO_PASSIVE

```
#define             LASSO_SAML2_STATUS_CODE_NO_PASSIVE
```

LASSO_SAML2_STATUS_CODE_NO_SUPPORTED_IDP

```
#define             LASSO_SAML2_STATUS_CODE_NO_SUPPORTED_IDP
```

LASSO_SAML2_STATUS_CODE_PARTIAL_LOGOUT

```
#define             LASSO_SAML2_STATUS_CODE_PARTIAL_LOGOUT
```

LASSO_SAML2_STATUS_CODE_PROXY_COUNT_EXCEEDED

```
#define             LASSO_SAML2_STATUS_CODE_PROXY_COUNT_EXCEEDED
```

LASSO_SAML2_STATUS_CODE_REQUEST_DENIED

```
#define             LASSO_SAML2_STATUS_CODE_REQUEST_DENIED
```

LASSO_SAML2_STATUS_CODE_REQUEST_UNSUPPORTED

```
#define             LASSO_SAML2_STATUS_CODE_REQUEST_UNSUPPORTED
```

LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_DEPRECATED

```
#define             LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_DEPRECATED
```

LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_TOO_HIGH

```
#define             LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_TOO_HIGH
```

LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_TOO_LOW

```
#define             LASSO_SAML2_STATUS_CODE_REQUEST_VERSION_TOO_LOW
```

LASSO_SAML2_STATUS_CODE_RESOURCE_NOT_RECOGNIZED

```
#define             LASSO_SAML2_STATUS_CODE_RESOURCE_NOT_RECOGNIZED
```

LASSO_SAML2_STATUS_CODE_TOO_MANY_RESPONSES

```
#define             LASSO_SAML2_STATUS_CODE_TOO_MANY_RESPONSES
```

LASSO_SAML2_STATUS_CODE_UNKNOWN_ATTR_PROFILE

```
#define             LASSO_SAML2_STATUS_CODE_UNKNOWN_ATTR_PROFILE
```

LASSO_SAML2_STATUS_CODE_UNKNOWN_PRINCIPAL

```
#define             LASSO_SAML2_STATUS_CODE_UNKNOWN_PRINCIPAL
```

LASSO_SAML2_STATUS_CODE_UNSUPPORTED_BINDING

```
#define             LASSO_SAML2_STATUS_CODE_UNSUPPORTED_BINDING
```

LASSO_SAML2_AUTHN_CONTEXT_AUTHENTICATED_TELEPHONY

```
#define             LASSO_SAML2_AUTHN_CONTEXT_AUTHENTICATED_TELEPHONY
```

LASSO_SAML2_AUTHN_CONTEXT_INTERNET_PROTOCOL

```
#define             LASSO_SAML2_AUTHN_CONTEXT_INTERNET_PROTOCOL
```

LASSO_SAML2_AUTHN_CONTEXT_INTERNET_PROTOCOL_PASSWORD

```
#define             LASSO_SAML2_AUTHN_CONTEXT_INTERNET_PROTOCOL_PASSWORD
```

LASSO_SAML2_AUTHN_CONTEXT_KERBEROS

```
#define             LASSO_SAML2_AUTHN_CONTEXT_KERBEROS
```

LASSO_SAML2_AUTHN_CONTEXT_MOBILE_ONE_FACTOR_CONTRACT

```
#define             LASSO_SAML2_AUTHN_CONTEXT_MOBILE_ONE_FACTOR_CONTRACT
```

LASSO_SAML2_AUTHN_CONTEXT_MOBILE_ONE_FACTOR_UNREGISTERED

```
#define             LASSO_SAML2_AUTHN_CONTEXT_MOBILE_ONE_FACTOR_UNREGISTERED
```

LASSO_SAML2_AUTHN_CONTEXT_MOBILE_TWO_FACTOR_CONTRACT

```
#define             LASSO_SAML2_AUTHN_CONTEXT_MOBILE_TWO_FACTOR_CONTRACT
```

LASSO_SAML2_AUTHN_CONTEXT_MOBILE_TWO_FACTOR_UNREGISTERED

```
#define             LASSO_SAML2_AUTHN_CONTEXT_MOBILE_TWO_FACTOR_UNREGISTERED
```

LASSO_SAML2_AUTHN_CONTEXT_NOMAD_TELEPHONY

```
#define             LASSO_SAML2_AUTHN_CONTEXT_NOMAD_TELEPHONY
```

LASSO_SAML2_AUTHN_CONTEXT_PERSONALIZED_TELEPHONY

```
#define             LASSO_SAML2_AUTHN_CONTEXT_PERSONALIZED_TELEPHONY
```

LASSO_SAML2_AUTHN_CONTEXT_PGP

```
#define             LASSO_SAML2_AUTHN_CONTEXT_PGP
```

LASSO_SAML2_AUTHN_CONTEXT_PASSWORD_PROTECTED_TRANSPORT

```
#define             LASSO_SAML2_AUTHN_CONTEXT_PASSWORD_PROTECTED_TRANSPORT
```

LASSO_SAML2_AUTHN_CONTEXT_PASSWORD

```
#define             LASSO_SAML2_AUTHN_CONTEXT_PASSWORD
```

LASSO_SAML2_AUTHN_CONTEXT_PREVIOUS_SESSION

```
#define             LASSO_SAML2_AUTHN_CONTEXT_PREVIOUS_SESSION
```

LASSO_SAML2_AUTHN_CONTEXT_SMARTCARD

```
#define             LASSO_SAML2_AUTHN_CONTEXT_SMARTCARD
```

LASSO_SAML2_AUTHN_CONTEXT_SMARTCARD_PKI

```
#define             LASSO_SAML2_AUTHN_CONTEXT_SMARTCARD_PKI
```

LASSO_SAML2_AUTHN_CONTEXT_SOFTWARE_PKI

```
#define             LASSO_SAML2_AUTHN_CONTEXT_SOFTWARE_PKI
```

LASSO_SAML2_AUTHN_CONTEXT_SPKI

```
#define             LASSO_SAML2_AUTHN_CONTEXT_SPKI
```

LASSO_SAML2_AUTHN_CONTEXT_SECURE_REMOTE_PASSWORD

```
#define             LASSO_SAML2_AUTHN_CONTEXT_SECURE_REMOTE_PASSWORD
```

LASSO_SAML2_AUTHN_CONTEXT_TLS_CLIENT

```
#define             LASSO_SAML2_AUTHN_CONTEXT_TLS_CLIENT
```

LASSO_SAML2_AUTHN_CONTEXT_X509

```
#define             LASSO_SAML2_AUTHN_CONTEXT_X509
```

LASSO_SAML2_AUTHN_CONTEXT_TELEPHONY

```
#define             LASSO_SAML2_AUTHN_CONTEXT_TELEPHONY
```

LASSO_SAML2_AUTHN_CONTEXT_TIME_SYNC_TOKEN

```
#define             LASSO_SAML2_AUTHN_CONTEXT_TIME_SYNC_TOKEN
```

LASSO_SAML2_AUTHN_CONTEXT_XMLDSIG

```
#define             LASSO_SAML2_AUTHN_CONTEXT_XMLDSIG
```

LASSO_SAML2_AUTHN_CONTEXT_UNSPECIFIED

```
#define             LASSO_SAML2_AUTHN_CONTEXT_UNSPECIFIED
```

LASSO_SAML2_CONFIRMATION_METHOD_BEARER

```
#define LASSO_SAML2_CONFIRMATION_METHOD_BEARER "urn:oasis:names:tc:SAML:2.0:cm:bearer"
```

LASSO_SAML2_CONFIRMATION_METHOD_HOLDER_OF_KEY

```
#define LASSO_SAML2_CONFIRMATION_METHOD_HOLDER_OF_KEY "urn:oasis:names:tc:SAML:2.0:cm: ←
    holder-of-key"
```

LASSO_SAML2_FIELD_RESPONSE

```
#define LASSO_SAML2_FIELD_RESPONSE "SAMLResponse"
```

LASSO_SAML2_FIELD_REQUEST

```
#define LASSO_SAML2_FIELD_REQUEST "SAMLRequest"
```

LASSO_SAML2_FIELD_ARTIFACT

```
#define LASSO_SAML2_FIELD_ARTIFACT "SAMLart"
```

LASSO_SAML2_FIELD_RELAYSTATE

```
#define LASSO_SAML2_FIELD_RELAYSTATE "RelayState"
```

LASSO_SAML2_FIELD_SIGNATURE

```
#define LASSO_SAML2_FIELD_SIGNATURE "Signature"
```

LASSO_SAML2_FIELD_SIGALG

```
#define LASSO_SAML2_FIELD_SIGALG "SigAlg"
```

See Also

LassoSamlp2AuthnRequest, LassoSaml2Assertion, LassoLogin

7.2 LassoSaml2Action

LassoSaml2Action — <saml2:Action>

Functions

LassoNode *	lasso_saml2_action_new ()
LassoNode *	lasso_saml2_action_new_with_string ()

Types and Values

struct	LassoSaml2Action

Description

```
<complexType name="ActionType">
  <simpleContent>
    <extension base="string">
      <attribute name="Namespace" type="anyURI" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

Figure 7.1: Schema fragment for saml2:Action

Functions

lasso_saml2_action_new ()

```
LassoNode~*
lasso_saml2_action_new (void);
```

Creates a new LassoSaml2Action object.

Returns

a newly created LassoSaml2Action object

lasso_saml2_action_new_with_string ()

```
LassoNode~*
lasso_saml2_action_new_with_string (char *content);
```

Creates a new LassoSaml2Action object and initializes it with `content` .

Parameters

content	content of the new element

Returns

a newly created LassoSaml2Action object

Types and Values

struct LassoSaml2Action

```
struct LassoSaml2Action {
  LassoNode parent;

  /* elements */
  char *content;
  /* attributes */
  char *Namespace;
};
```

7.3 LassoSaml2Advice

LassoSaml2Advice — <saml2:Advice>

Functions

LassoNode *	lasso_saml2_advice_new ()

Types and Values

struct	LassoSaml2Advice

Description

```
<complexType name="AdviceType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="saml:AssertionIDRef"/>
    <element ref="saml:AssertionURIRef"/>
    <element ref="saml:Assertion"/>
    <element ref="saml:EncryptedAssertion"/>
    <any namespace="##other" processContents="lax"/>
  </choice>
</complexType>
```

Figure 7.2: Schema fragment for saml2:Advice

Functions

lasso_saml2_advice_new ()

```
LassoNode~*
lasso_saml2_advice_new (void);
```

Creates a new LassoSaml2Advice object.

Returns

a newly created LassoSaml2Advice object

Types and Values

struct LassoSaml2Advice

```
struct LassoSaml2Advice {
  LassoNode parent;

  /* elements */
  GList *AssertionIDRef; /* of LassoNode */
  GList *AssertionURIRef; /* of LassoNode */
  GList *Assertion; /* of LassoSaml2Assertion */
  GList *EncryptedAssertion; /* of LassoSaml2EncryptedElement */
};
```

7.4 LassoSaml2Assertion

LassoSaml2Assertion — <saml2:Assertion>

Functions

LassoNode *	lasso_saml2_assertion_new ()
gboolean	lasso_saml2_assertion_has_audience_restriction ()
gboolean	lasso_saml2_assertion_is_audience_restricted ()
void	lasso_saml2_assertion_set_subject_name_id ()
void	lasso_saml2_assertion_set_subject_confirmation_name_id ()
LassoSaml2SubjectConfirmationData *	lasso_saml2_assertion_get_subject_confirmation_data ()
void	lasso_saml2_assertion_set_subject_confirmation_data ()
void	lasso_saml2_assertion_set_basic_conditions ()
void	lasso_saml2_assertion_add_audience_restriction ()
void	lasso_saml2_assertion_add_proxy_limit ()
LassoSaml2AssertionValidationState	lasso_saml2_assertion_validate_conditions ()
LassoSaml2AssertionValidationState	lasso_saml2_assertion_validate_time_checks ()
LassoSaml2AssertionValidationState	lasso_saml2_assertion_validate_audience ()
LassoSaml2AssertionValidationState	lasso_saml2_assertion_allows_proxying ()
LassoSaml2AssertionValidationState	lasso_saml2_assertion_allows_proxying_to ()
LassoProvider *	lasso_saml2_assertion_get_issuer_provider ()
lasso_error_t	lasso_saml2_assertion_add_attribute_with_node ()
const char *	lasso_saml2_assertion_get_in_response_to ()
lasso_error_t	lasso_saml2_assertion_decrypt_subject ()
gboolean	lasso_saml2_assertion_has_one_time_use ()
void	lasso_saml2_assertion_set_one_time_use ()

Types and Values

struct	LassoSaml2Assertion
enum	LassoSaml2AssertionValidationState
#define	LASSO_DURATION_DAY
#define	LASSO_DURATION_HOUR
#define	LASSO_DURATION_MINUTE
#define	LASSO_DURATION_WEEK

Description

```
<complexType name="AssertionType">
  <sequence>
    <element ref="saml:Issuer"/>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="saml:Subject" minOccurs="0"/>
    <element ref="saml:Conditions" minOccurs="0"/>
    <element ref="saml:Advice" minOccurs="0"/>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="saml:Statement"/>
      <element ref="saml:AuthnStatement"/>
      <element ref="saml:AuthzDecisionStatement"/>
      <element ref="saml:AttributeStatement"/>
    </choice>
  </sequence>
  <attribute name="Version" type="string" use="required"/>
  <attribute name="ID" type="ID" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
</complexType>
```

Figure 7.3: Schema fragment for saml2:Assertion

Functions

lasso_saml2_assertion_new ()

```
LassoNode~*
lasso_saml2_assertion_new (void);
```

Creates a new LassoSaml2Assertion object.

Returns

a newly created LassoSaml2Assertion object

lasso_saml2_assertion_has_audience_restriction ()

```
gboolean
lasso_saml2_assertion_has_audience_restriction
                               (LassoSaml2Assertion *saml2_assertion);
```

Verify that a LassoSaml2AudienceRestriction is present in the assertion.

Parameters

saml2_assertion	a LassoSaml2Assertion object

Returns

TRUE if a LassoSaml2AudienceRestriction is present in the Conditions of the LassoSaml2Assertion.

lasso_saml2_assertion_is_audience_restricted ()

```
gboolean
lasso_saml2_assertion_is_audience_restricted
                               (LassoSaml2Assertion *saml2_assertion,
                                char *providerID);
```

Verify that the assertion is restricted to the given providerID.

Parameters

saml2_assertion	a LassoSaml2Assertion object
providerID	the providerID that will be compared to the audience restriction declarations.

Returns

TRUE if *providerID* is part of a LassoSaml2AudienceRestriction element in the assertion, FALSE otherwise.

lasso_saml2_assertion_set_subject_name_id ()

```
void
lasso_saml2_assertion_set_subject_name_id
                               (LassoSaml2Assertion *saml2_assertion,
                                LassoNode *node);
```

Set the subject NameID, which can be a simple LassoSaml2NameID object or an encrypted LassoSaml2NameID as a LassoSaml2EncryptedElement.

Parameters

saml2_assertion	a LassoSaml2Assertion object
node	a LassoSaml2NameID or LassoSaml2EncryptedElement

lasso_saml2_assertion_set_subject_confirmation_name_id ()

```
void
lasso_saml2_assertion_set_subject_confirmation_name_id
                               (LassoSaml2Assertion *saml2_assertion,
                                LassoNode *node);
```

Set the subject NameID, which can be a simple LassoSaml2NameID object or an encrypted LassoSaml2NameID as a LassoSaml2EncryptedElement.

Parameters

saml2_assertion	a LassoSaml2Assertion object	
node	a LassoSaml2NameID or LassoSaml2EncryptedElement	

lasso_saml2_assertion_get_subject_confirmation_data ()

```
LassoSaml2SubjectConfirmationData~*
lasso_saml2_assertion_get_subject_confirmation_data
                               (LassoSaml2Assertion *saml2_assertion,
                                gboolean create);
```

lasso_saml2_assertion_set_subject_confirmation_data ()

```
void
lasso_saml2_assertion_set_subject_confirmation_data
                               (LassoSaml2Assertion *saml2_assertion,
                                time_t tolerance,
                                time_t length,
                                const char *Recipient,
                                const char *InResponseTo,
                                const char *Address);
```

lasso_saml2_assertion_set_basic_conditions ()

```
void
lasso_saml2_assertion_set_basic_conditions
                               (LassoSaml2Assertion *saml2_assertion,
                                time_t tolerance,
                                time_t length,
                                gboolean one_time_use);
```

Set conditions limiting usage of the assertion.

`tolerance` and `length` are time quantity measured in seconds, it defines the time range in which the assertion is valid, it is computed as [now()-tolerance, now()+length+tolerance]. `one_time_use` allows the issuer to limit caching of the assertion. `proxy_count` specify how many proxy hop can be traversed before this assertion should lose any trust.

Parameters

tolerance	tolerance to the range of time when the assertion is valid.	*[default -1]*
length	length of the range of time when the assertion is valid.	*[default -1]*
one_time_use	can the assertion be kept or should it be used immediately.	*[default FALSE]*

lasso_saml2_assertion_add_audience_restriction ()

```
void
lasso_saml2_assertion_add_audience_restriction
                               (LassoSaml2Assertion *saml2_assertion,
                                const char *providerID);
```

Add an audience restriction to a LassoSaml2Assertion.

Parameters

saml2_assertion	a LassoSaml2Assertion object	
providerId	the provider id to restrict audience to	

lasso_saml2_assertion_add_proxy_limit ()

```
void
lasso_saml2_assertion_add_proxy_limit (LassoSaml2Assertion *saml2_assertion,
                                       int proxy_count,
                                       GList *proxy_audiences);
```

A LassoSaml2ProxyRestriction to the conditions of consumption of `saml2_assertion` .

Parameters

saml2_assertion	a LassoSaml2Assertion object	
proxy_count	the number of hops in the proxy chain, a negative value means no limitation.	*[default -1]*
proxy_audiences	a list of audience restriction for newly issued assertion based on the `saml2_assertion` assertion. An empty list means no audience restriction.	*[allow-none][element-type string]*

lasso_saml2_assertion_validate_conditions ()

```
LassoSaml2AssertionValidationState
lasso_saml2_assertion_validate_conditions
                               (LassoSaml2Assertion *saml2_assertion,
                                const char *relaying_party_providerID);
```

Check the validation of the assertion with respect to the conditions of consumption that it contains. System functions are used for getting current time and checking eventual time constraints.

Parameters

saml2_assertion	a LassoSaml2Assertion object	
relaying_party_providerID	the providerID of the current relaying party, use to check for audience restrictions.	*[allow-none]*

Returns

LASSO_SAML2_ASSERTION_VALID if the assertion is valid, LASSO_SAML2_ASSERTION_INVALID is some check failed, LASSO_SAML2_ASSERTION_INDETERMINATE if somehting was impossible to eveluate.

lasso_saml2_assertion_validate_time_checks ()

```
LassoSaml2AssertionValidationState
lasso_saml2_assertion_validate_time_checks
                               (LassoSaml2Assertion *saml2_assertion,
                                unsigned int tolerance,
                                time_t now);
```

Check if the `saml2_assertion` conditions about NotBefore and NotOnOrAfter are valid with respect to the `now` time or the current time. `tolerance` allows to loosely check for validatity, i.e. start time is decreased of `tolerance` seconds and end time is increased of `tolerance` seconds.

Parameters

saml2_assertion	a LassoSaml2Assertion object	
tolerance	a duration as seconds	
now	the current time as seconds since EPOCH or 0 to use the system time.	*[default 0]*

Returns

a value among LassoSaml2AssertionValidationState.

lasso_saml2_assertion_validate_audience ()

```
LassoSaml2AssertionValidationState
lasso_saml2_assertion_validate_audience
                               (LassoSaml2Assertion *saml2_assertion,
                                const gchar *audience);
```

Check if the `saml2_assertion` is directed to a given `audience` .

Parameters

saml2_assertion	a LassoSaml2Assertion object	
audience	the name of an entity	

Returns

a value among LassoSaml2AssertionValidationState enumeration.

lasso_saml2_assertion_allows_proxying ()

```
LassoSaml2AssertionValidationState
lasso_saml2_assertion_allows_proxying (LassoSaml2Assertion *saml2_assertion);
```

Test whether this `saml2_assertion` allows to mint new assertion on the basis of it.

It verifies that the proxying count is positive (or absent).

Parameters

saml2_assertion	a LassoSaml2Assertion object	

Returns

a value among LassoSaml2AssertionValidationState enumeration. LASSO_SAML2_ASSERTION_INDETERMINATE usually means that an element was not conform to the XML Schema for SAML 2.0.

lasso_saml2_assertion_allows_proxying_to ()

```
LassoSaml2AssertionValidationState
lasso_saml2_assertion_allows_proxying_to
                               (LassoSaml2Assertion *saml2_assertion,
                                const char *audience);
```

Test whether this `saml2_assertion` allows to mint new assertion on the basis of it targetted for `audience`.

It verifies that if `audience` is non-NULL it is part of the proxy Audience restriction. If `audience` is NULL, it checks that no proxying Audience restriction is present.

Parameters

saml2_assertion	a LassoSaml2Assertion object	
audience	the relaying party which we want to proxy to.	*[allow-none]*

Returns

a value among LassoSaml2AssertionValidationState enumeration. LASSO_SAML2_ASSERTION_INDETERMINATE usually means that an element was not conform to the XML Schema for SAML 2.0.

lasso_saml2_assertion_get_issuer_provider ()

```
LassoProvider~*
lasso_saml2_assertion_get_issuer_provider
                               (const LassoSaml2Assertion *saml2_assertion,
                                const LassoServer *server);
```

Return the LassoProvider object for the provider who created this assertion.

Parameters

saml2_assertion	a LassoSaml2 assertion	
server	a LassoServer object	

Returns

a LassoProvider object, or NULL if the Issuer element is missing, or the given provider unknown to the LassoServer object.

lasso_saml2_assertion_add_attribute_with_node ()

```
lasso_error_t
lasso_saml2_assertion_add_attribute_with_node
                               (LassoSaml2Assertion *assertion,
                                const char *name,
                                const char *nameformat,
                                LassoNode *content);
```

Add a new attribute declaration and set this node as the content.

Parameters

assertion	a LassoSaml2Assertion object	
name	the attribute name	
name_format	the attribute name format (the namespace of the name)	
content	a LassoNode object to put as content of the attribute	

Returns

0 if successful, an error code otherwise.

lasso_saml2_assertion_get_in_response_to ()

```
const char~*
lasso_saml2_assertion_get_in_response_to
                               (LassoSaml2Assertion *assertion);
```

Return the ID of the request this assertion respond to.

Parameters

assertion	a LassoSaml2Assertion object	

Returns

the InResponseTo attribute content of the SubjectConfirmationData if found

lasso_saml2_assertion_decrypt_subject ()

```
lasso_error_t
lasso_saml2_assertion_decrypt_subject (LassoSaml2Assertion *assertion,
                                       LassoServer *server);
```

Decipher (if needed) the EncryptedID of the Subject.

Parameters

assertion	a LassoSaml2Assertion object
server	a LassoServer object

Returns

0 if successful, an error code otherwise. See lasso_saml2_encrypted_element_server_decrypt().

lasso_saml2_assertion_has_one_time_use ()

```
gboolean
lasso_saml2_assertion_has_one_time_use
                               (LassoSaml2Assertion *saml2_assertion);
```

Return whether this assertion has the OneTimeUse property.

In this case the relaying party must add the assertion ID to a OneTimeUser cache and discards any assertion received in the future with the same ID.

Parameters

saml2_assertion	a LassoSaml2Assertion object

Returns

TRUE if this assertion has the property OneTimeUse, FALSE otherwise.

lasso_saml2_assertion_set_one_time_use ()

```
void
lasso_saml2_assertion_set_one_time_use
                               (LassoSaml2Assertion *saml2_assertion,
                                gboolean one_time_use);
```

Set the one time use condition on this assertion.

Parameters

saml2_assertion	a LassoSaml2Assertion object	
one_time_use	is this assertion to be used one time only ?	

Types and Values

struct LassoSaml2Assertion

```
struct LassoSaml2Assertion {
  LassoNode parent;

  /* elements */
  LassoSaml2NameID *Issuer;
  LassoSaml2Subject *Subject;
  LassoSaml2Conditions *Conditions;
  LassoSaml2Advice *Advice;
  GList *Statement; /* of LassoSaml2StatementAbstract */
  GList *AuthnStatement; /* of LassoSaml2AuthnStatement */
  GList *AuthzDecisionStatement; /* of LassoSaml2AuthzDecisionStatement */
  GList *AttributeStatement; /* of LassoSaml2AttributeStatement */
  /* attributes */
  char *Version;
  char *ID;
  char *IssueInstant;
};
```

enum LassoSaml2AssertionValidationState

Members

LASSO_SAML2_ASSERTION_VALID		
LASSO_SAML2_ASSERTION_INVALID		
LASSO_SAML2_ASSERTION_INDETERMINATE		

LASSO_DURATION_DAY

```
#define LASSO_DURATION_DAY 24*LASSO_DURATION_HOUR
```

Number of seconds in a day.

LASSO_DURATION_HOUR

```
#define LASSO_DURATION_HOUR 3600
```

Number of seconds in a hour.

LASSO_DURATION_MINUTE

```
#define LASSO_DURATION_MINUTE 60
```

Number of seconds in a minute.

LASSO_DURATION_WEEK

```
#define LASSO_DURATION_WEEK 7*LASSO_DURATION_DAY
```

Number of seconds in a week.

7.5 LassoSaml2AttributeStatement

LassoSaml2AttributeStatement — <saml2:AttributeStatement>

Functions

LassoNode *	lasso_saml2_attribute_statement_new ()

Types and Values

struct	LassoSaml2AttributeStatement

Description

```
<complexType name="AttributeStatementType">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <choice maxOccurs="unbounded">
        <element ref="saml:Attribute"/>
        <element ref="saml:EncryptedAttribute"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.4: Schema fragment for saml2:AttributeStatement

Functions

lasso_saml2_attribute_statement_new ()

```
LassoNode~*
lasso_saml2_attribute_statement_new (void);
```

Creates a new LassoSaml2AttributeStatement object.

Returns

a newly created LassoSaml2AttributeStatement object

Types and Values

struct LassoSaml2AttributeStatement

```
struct LassoSaml2AttributeStatement {
  LassoSaml2StatementAbstract parent;

  /* elements */
  GList *Attribute; /* of LassoSaml2Attribute */
  GList *EncryptedAttribute; /* of LassoSaml2EncryptedElement */
};
```

7.6 LassoSaml2AttributeValue

LassoSaml2AttributeValue — value of an attribute in a SAML 2.0 assertion

Functions

LassoSaml2AttributeValue *	lasso_saml2_attribute_value_new ()

Types and Values

struct	LassoSaml2AttributeValue

Description

```
<element name="AttributeValue" type="anyType" nillable="true"/>
```

Figure 7.5: Schema fragment from saml-schema-assertion-2.0.xsd)

This object support a special of specifying its content. If the `any` attribute is NULL, then you can attach an xmlNode using lasso_node_set_original_xmlnode() and it will be used to generate the content of the serialized xmlNode for this object. The content (attributes, childrent and namespaces) of the

node will be copied to the result node created by a call to lasso_node_get_xmlNode().

Functions

lasso_saml2_attribute_value_new ()

```
LassoSaml2AttributeValue~*
lasso_saml2_attribute_value_new (void);
```

Types and Values

struct LassoSaml2AttributeValue

```
struct LassoSaml2AttributeValue {
  LassoNode parent;

  GList *any; /* of LassoNode */
};
```

See Also

LassoSaml2Attribute, LassoSaml2AttributeStatement, LassoSaml2Assertion

7.7 LassoSaml2Attribute

LassoSaml2Attribute — <saml2:Attribute>

Functions

LassoNode *	lasso_saml2_attribute_new ()

Types and Values

struct	LassoSaml2Attribute

Description

```
<complexType name="AttributeType">
  <sequence>
    <element ref="saml:AttributeValue" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Name" type="string" use="required"/>
  <attribute name="NameFormat" type="anyURI" use="optional"/>
  <attribute name="FriendlyName" type="string" use="optional"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

Figure 7.6: Schema fragment for saml2:Attribute

Functions

lasso_saml2_attribute_new ()

```
LassoNode~*
lasso_saml2_attribute_new (void);
```

Creates a new LassoSaml2Attribute object.

Returns

a newly created LassoSaml2Attribute object

Types and Values

struct LassoSaml2Attribute

```
struct LassoSaml2Attribute {
  LassoNode parent;

  /* elements */
  GList *AttributeValue; /* of LassoNode */
  /* attributes */
  char *Name;
  char *NameFormat;
  char *FriendlyName;
};
```

7.8 LassoSaml2AudienceRestriction

LassoSaml2AudienceRestriction — <saml2:AudienceRestriction>

Functions

LassoNode *	lasso_saml2_audience_restriction_new ()

Types and Values

struct	LassoSaml2AudienceRestriction

Description

```
<complexType name="AudienceRestrictionType">
  <complexContent>
    <extension base="saml:ConditionAbstractType">
      <sequence>
        <element ref="saml:Audience" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.7: Schema fragment for saml2:AudienceRestriction

Functions

lasso_saml2_audience_restriction_new ()

```
LassoNode~*
lasso_saml2_audience_restriction_new (void);
```

Creates a new LassoSaml2AudienceRestriction object.

Returns

a newly created LassoSaml2AudienceRestriction object

Types and Values

struct LassoSaml2AudienceRestriction

```
struct LassoSaml2AudienceRestriction {
  LassoSaml2ConditionAbstract parent;

  /* elements */
  char *Audience;
};
```

7.9 LassoSaml2AuthnContext

LassoSaml2AuthnContext — <saml2:AuthnContext>

Functions

LassoNode *	lasso_saml2_authn_context_new ()

Types and Values

struct	LassoSaml2AuthnContext

Description

```
<complexType name="AuthnContextType">
  <sequence>
    <choice>
      <sequence>
        <element ref="saml:AuthnContextClassRef"/>
        <choice minOccurs="0">
          <element ref="saml:AuthnContextDecl"/>
          <element ref="saml:AuthnContextDeclRef"/>
        </choice>
      </sequence>
      <choice>
        <element ref="saml:AuthnContextDecl"/>
        <element ref="saml:AuthnContextDeclRef"/>
      </choice>
    </choice>
    <element ref="saml:AuthenticatingAuthority" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Figure 7.8: Schema fragment for saml2:AuthnContext

Functions

lasso_saml2_authn_context_new ()

```
LassoNode~*
lasso_saml2_authn_context_new (void);
```

Creates a new LassoSaml2AuthnContext object.

Returns

a newly created LassoSaml2AuthnContext object

Types and Values

struct LassoSaml2AuthnContext

```
struct LassoSaml2AuthnContext {
  LassoNode parent;

  /* elements */
  char *AuthnContextClassRef;
  /* XXX */ void *AuthnContextDecl;
  char *AuthnContextDeclRef;
  char *AuthenticatingAuthority;
};
```

7.10 LassoSaml2AuthnStatement

LassoSaml2AuthnStatement — <saml2:AuthnStatement>

Functions

LassoNode *	lasso_saml2_authn_statement_new ()

Types and Values

struct	LassoSaml2AuthnStatement

Description

```
<complexType name="AuthnStatementType">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <sequence>
        <element ref="saml:SubjectLocality" minOccurs="0"/>
        <element ref="saml:AuthnContext"/>
      </sequence>
      <attribute name="AuthnInstant" type="dateTime" use="required"/>
      <attribute name="SessionIndex" type="string" use="optional"/>
      <attribute name="SessionNotOnOrAfter" type="dateTime" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.9: Schema fragment for saml2:AuthnStatement

Functions

lasso_saml2_authn_statement_new ()

```
LassoNode~*
lasso_saml2_authn_statement_new (void);
```

Creates a new LassoSaml2AuthnStatement object.

Returns

a newly created LassoSaml2AuthnStatement object

Types and Values

struct LassoSaml2AuthnStatement

```
struct LassoSaml2AuthnStatement {
  LassoSaml2StatementAbstract parent;

  /* elements */
  LassoSaml2SubjectLocality *SubjectLocality;
  LassoSaml2AuthnContext *AuthnContext;
  /* attributes */
  char *AuthnInstant;
  char *SessionIndex;
  char *SessionNotOnOrAfter;
};
```

7.11 LassoSaml2AuthzDecisionStatement

LassoSaml2AuthzDecisionStatement — <saml2:AuthzDecisionStatement>

Functions

LassoNode *	lasso_saml2_authz_decision_statement_new ()

Types and Values

struct	LassoSaml2AuthzDecisionStatement

Description

```
<complexType name="AuthzDecisionStatementType">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <sequence>
        <element ref="saml:Action" maxOccurs="unbounded"/>
        <element ref="saml:Evidence" minOccurs="0"/>
      </sequence>
      <attribute name="Resource" type="anyURI" use="required"/>
      <attribute name="Decision" type="saml:DecisionType" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.10: Schema fragment for saml2:AuthzDecisionStatement

Functions

lasso_saml2_authz_decision_statement_new ()

```
LassoNode~*
lasso_saml2_authz_decision_statement_new
                                (void);
```

Creates a new LassoSaml2AuthzDecisionStatement object.

Returns

a newly created LassoSaml2AuthzDecisionStatement object

Types and Values

struct LassoSaml2AuthzDecisionStatement

```
struct LassoSaml2AuthzDecisionStatement {
  LassoSaml2StatementAbstract parent;

  /* elements */
  LassoSaml2Action *Action;
  LassoSaml2Evidence *Evidence;
  /* attributes */
  char *Resource;
  char *Decision;
};
```

7.12 LassoSaml2BaseIDAbstract

LassoSaml2BaseIDAbstract — <saml2:BaseIDAbstract>

Functions

LassoNode *	lasso_saml2_base_idabstract_new ()

Types and Values

struct	LassoSaml2BaseIDAbstract

Description

```
<complexType name="BaseIDAbstractType" abstract="true">
  <attributeGroup ref="saml:IDNameQualifiers"/>
</complexType>
```

Figure 7.11: Schema fragment for saml2:BaseIDAbstract

Functions

lasso_saml2_base_idabstract_new ()

```
LassoNode~*
lasso_saml2_base_idabstract_new (void);
```

Creates a new LassoSaml2BaseIDAbstract object.

Returns

a newly created LassoSaml2BaseIDAbstract object

Types and Values

struct LassoSaml2BaseIDAbstract

```
struct LassoSaml2BaseIDAbstract {
  LassoNode parent;

  /* attributes */
  char *NameQualifier;
  char *SPNameQualifier;
};
```

7.13 LassoSaml2ConditionAbstract

LassoSaml2ConditionAbstract — <saml2:ConditionAbstract>

Functions

LassoNode *	lasso_saml2_condition_abstract_new ()

Types and Values

struct	LassoSaml2ConditionAbstract

Description

```
<complexType name="ConditionAbstractType" abstract="true"/>
<element name="AudienceRestriction" type="saml:AudienceRestrictionType"/>
<complexType name="AudienceRestrictionType">
  <complexContent>
    <extension base="saml:ConditionAbstractType">
      <sequence>
        <element ref="saml:Audience" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.12: Schema fragment for saml2:ConditionAbstract

Functions

lasso_saml2_condition_abstract_new ()

```
LassoNode~*
lasso_saml2_condition_abstract_new (void);
```

Creates a new LassoSaml2ConditionAbstract object.

Returns

a newly created LassoSaml2ConditionAbstract object

Types and Values

struct LassoSaml2ConditionAbstract

```
struct LassoSaml2ConditionAbstract {
  LassoNode parent;
};
```

7.14 LassoSaml2Conditions

LassoSaml2Conditions — <saml2:Conditions>

Functions

LassoNode *	lasso_saml2_conditions_new ()

Types and Values

struct	LassoSaml2Conditions

Description

```
<complexType name="ConditionsType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="saml:Condition"/>
    <element ref="saml:AudienceRestriction"/>
    <element ref="saml:OneTimeUse"/>
    <element ref="saml:ProxyRestriction"/>
  </choice>
  <attribute name="NotBefore" type="dateTime" use="optional"/>
  <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
</complexType>
```

Figure 7.13: Schema fragment for saml2:Conditions

Functions

lasso_saml2_conditions_new ()

```
LassoNode~*
lasso_saml2_conditions_new (void);
```

Creates a new LassoSaml2Conditions object.

Returns

a newly created LassoSaml2Conditions object

Types and Values

struct LassoSaml2Conditions

```
struct LassoSaml2Conditions {
  LassoNode parent;

  /* elements */
  GList *Condition; /* of LassoSaml2ConditionAbstract */
  GList *AudienceRestriction; /* of LassoSaml2AudienceRestriction */
  GList *OneTimeUse; /* of LassoSaml2OneTimeUse */
  GList *ProxyRestriction; /* of LassoSaml2ProxyRestriction */
  /* attributes */
  char *NotBefore;
  char *NotOnOrAfter;
};
```

7.15 LassoSaml2EncryptedElement

LassoSaml2EncryptedElement — <saml2:EncryptedElement>

Functions

LassoNode *	lasso_saml2_encrypted_element_new ()
lasso_error_t	lasso_saml2_encrypted_element_decrypt ()
LassoSaml2EncryptedElement *	lasso_saml2_encrypted_element_build_encrypted_persistent_name_id ()
lasso_error_t	lasso_saml2_encrypted_element_server_decrypt ()

Types and Values

struct	LassoSaml2EncryptedElement

Description

This element can contain an encrypted XML document fragment, use lasso_saml2_encrypted_element_decrypt() to retrieve it.

```
<complexType name="EncryptedElementType">
  <sequence>
    <element ref="xenc:EncryptedData"/>
    <element ref="xenc:EncryptedKey" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Figure 7.14: Schema fragment for saml2:EncryptedElement

Functions

lasso_saml2_encrypted_element_new ()

```
LassoNode~*
lasso_saml2_encrypted_element_new (void);
```

Creates a new LassoSaml2EncryptedElement object.

Returns

a newly created LassoSaml2EncryptedElement object

lasso_saml2_encrypted_element_decrypt ()

```
lasso_error_t
lasso_saml2_encrypted_element_decrypt (LassoSaml2EncryptedElement *encrypted_element,
                                       xmlSecKey *encryption_private_key,
                                       LassoNode **decrypted_node);
```

Decrypt the content of a LassoSaml2EncryptedElement using the given xmlSecKey. The xmlNode resulting of decrypting it is converted into a LassoNode object.

Parameters

encrypted_element	the LassoSaml2EncryptedElement to decrypt	
encryption_private_key	the xmlSecKey to decrypt the node	

Returns

0 if successful, an error otherwise.

lasso_saml2_encrypted_element_build_encrypted_persistent_name_id ()

```
LassoSaml2EncryptedElement~*
lasso_saml2_encrypted_element_build_encrypted_persistent_name_id
                              (const char *id,
                               const char *idpID,
                               const LassoProvider *provider);
```

lasso_saml2_encrypted_element_server_decrypt ()

```
lasso_error_t
lasso_saml2_encrypted_element_server_decrypt
                               (LassoSaml2EncryptedElement *encrypted_element,
                                LassoServer *server,
                                LassoNode **decrypted_node);
```

Decrypt the given encrypted element using the encryption private key of the `server` object

Parameters

encrypted_element	a LassoSaml2EncryptedElement object	
server	a LassoServer object	
decrypted_node	an output arg for a LassoNode.	*[out]*

Returns

0 if successful, an error code otherwise. See lasso_saml2_encrypted_element_server_decrypt().

Types and Values

struct LassoSaml2EncryptedElement

```
struct LassoSaml2EncryptedElement {
  LassoNode parent;

  /* elements */
  xmlNode *EncryptedData;
  GList *EncryptedKey; /* of xmlNode* */
  LassoNode *original_data;
};
```

7.16 LassoSaml2Evidence

LassoSaml2Evidence — <saml2:Evidence>

Functions

LassoNode *	lasso_saml2_evidence_new ()

Types and Values

struct	LassoSaml2Evidence

Description

```
<complexType name="EvidenceType">
  <choice maxOccurs="unbounded">
    <element ref="saml:AssertionIDRef"/>
    <element ref="saml:AssertionURIRef"/>
    <element ref="saml:Assertion"/>
    <element ref="saml:EncryptedAssertion"/>
  </choice>
</complexType>
```

Figure 7.15: Schema fragment for saml2:Evidence

Functions

lasso_saml2_evidence_new ()

```
LassoNode~*
lasso_saml2_evidence_new (void);
```

Creates a new LassoSaml2Evidence object.

Returns

a newly created LassoSaml2Evidence object

Types and Values

struct LassoSaml2Evidence

```
struct LassoSaml2Evidence {
  LassoNode parent;

  /* elements */
  GList *AssertionIDRef; /* of LassoNode */
  GList *AssertionURIRef; /* of LassoNode */
  GList *Assertion; /* of LassoSaml2Assertion */
  GList *EncryptedAssertion; /* of LassoSaml2EncryptedElement */
};
```

7.17 LassoSaml2KeyInfoConfirmationData

LassoSaml2KeyInfoConfirmationData — <saml2:KeyInfoConfirmationData>

Functions

LassoNode *	lasso_saml2_key_info_confirmation_data_new ()

Types and Values

struct	LassoSaml2KeyInfoConfirmationData

Description

```
<complexType name="KeyInfoConfirmationDataType" mixed="false">
  <complexContent>
    <restriction base="saml:SubjectConfirmationDataType">
      <sequence>
        <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

Figure 7.16: Schema fragment for saml2:KeyInfoConfirmationData

Functions

lasso_saml2_key_info_confirmation_data_new ()

```
LassoNode~*
lasso_saml2_key_info_confirmation_data_new
                               (void);
```

Creates a new LassoSaml2KeyInfoConfirmationData object.

Returns

a newly created LassoSaml2KeyInfoConfirmationData object

Types and Values

struct LassoSaml2KeyInfoConfirmationData

```
struct LassoSaml2KeyInfoConfirmationData {
  LassoNode parent;

  /* elements */
  /* XXX */ void *KeyInfo;
};
```

7.18 LassoSaml2NameID

LassoSaml2NameID — <saml2:NameID>

Functions

LassoNode *	lasso_saml2_name_id_new ()
LassoNode *	lasso_saml2_name_id_new_with_string ()
LassoSaml2NameID *	lasso_saml2_name_id_build_persistent ()
gboolean	lasso_saml2_name_id_equals ()
LassoSaml2NameID *	lasso_saml2_name_id_new_with_persistent_format ()

Types and Values

struct	LassoSaml2NameID

Description

```
<complexType name="NameIDType">
  <simpleContent>
    <extension base="string">
      <attributeGroup ref="saml:IDNameQualifiers"/>
      <attribute name="Format" type="anyURI" use="optional"/>
      <attribute name="SPProvidedID" type="string" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

Figure 7.17: Schema fragment for saml2:NameID

Functions

lasso_saml2_name_id_new ()

```
LassoNode~*
lasso_saml2_name_id_new (void);
```

Creates a new LassoSaml2NameID object.

Returns

a newly created LassoSaml2NameID object

lasso_saml2_name_id_new_with_string ()

```
LassoNode~*
lasso_saml2_name_id_new_with_string (char *content);
```

Creates a new LassoSaml2NameID object and initializes it with `content` . Beware that no format is set.

Parameters

content	the Name Identifier.

Returns

a newly created LassoSaml2NameID object

lasso_saml2_name_id_build_persistent ()

```
LassoSaml2NameID~*
lasso_saml2_name_id_build_persistent (const char *id,
                                      const char *idpID,
                                      const char *providerID);
```

Warning

`lasso_saml2_name_id_build_persistent` has been deprecated since version 2.3 and should not be used in newly-written code.

use lasso_saml2_name_id_new_with_persistent_format() instead.

Create a new LassoSaml2NameID object, which the LASSO_SAML2_NAME_IDENTIFIER_FORMAT_PERSISTENT format, *id* as content, *idpID* as NameQualifier and *providerID* as SPNameQualifier.

Parameters

id	the identifier for the principal	
idpID	the entity ID of the IdP	
providerID	the entity ID of the provider	

Returns

a newly created LassoSaml2NameID

lasso_saml2_name_id_equals ()

```
gboolean
lasso_saml2_name_id_equals (LassoSaml2NameID *name_id,
                            LassoSaml2NameID *other_name_id);
```

Return TRUE if *name_id* equals *other_name_id* .

Parameters

name_id	a LassoSaml2NameID object	
other_name_id	another LassoSaml2NameID object	

Returns

TRUE if the two NameID are equal and are LassoSaml2NameID objects, FALSE otherwise.

lasso_saml2_name_id_new_with_persistent_format ()

```
LassoSaml2NameID~*
lasso_saml2_name_id_new_with_persistent_format
                              (const char *id,
                               const char *idpID,
                               const char *providerID);
```

Create a new LassoSaml2NameID object, which the LASSO_SAML2_NAME_IDENTIFIER_FORMAT_PERSISTENT format, *id* as content, *idpID* as NameQualifier and *providerID* as SPNameQualifier.

Parameters

id	the identifier for the princiapl	
idpID	the entity ID of the IdP	
providerID	the entity ID of the provider	

Returns

a newly created LassoSaml2NameID

Since: 2.3

Types and Values

struct LassoSaml2NameID

```
struct LassoSaml2NameID {
  LassoNode parent;

  /* elements */
  char *content;
  /* attributes */
  char *Format;
  char *SPProvidedID;
  char *NameQualifier;
  char *SPNameQualifier;
};
```

7.19 LassoSaml2OneTimeUse

LassoSaml2OneTimeUse — <saml2:OneTimeUse>

Functions

LassoNode *	lasso_saml2_one_time_use_new ()

Types and Values

struct	LassoSaml2OneTimeUse

Description

```
<complexType name="OneTimeUseType">
  <complexContent>
    <extension base="saml:ConditionAbstractType"/>
  </complexContent>
</complexType>
```

Figure 7.18: Schema fragment for saml2:OneTimeUse

Functions

lasso_saml2_one_time_use_new ()

```
LassoNode~*
lasso_saml2_one_time_use_new (void);
```

Creates a new LassoSaml2OneTimeUse object.

Returns

a newly created LassoSaml2OneTimeUse object

Types and Values

struct LassoSaml2OneTimeUse

```
struct LassoSaml2OneTimeUse {
  LassoSaml2ConditionAbstract parent;
};
```

7.20 LassoSaml2ProxyRestriction

LassoSaml2ProxyRestriction — <saml2:ProxyRestriction>

Functions

LassoNode *	lasso_saml2_proxy_restriction_new ()

Types and Values

struct	LassoSaml2ProxyRestriction

Description

```
<complexType name="ProxyRestrictionType">
  <complexContent>
    <extension base="saml:ConditionAbstractType">
      <sequence>
        <element ref="saml:Audience" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="Count" type="nonNegativeInteger" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.19: Schema fragment for saml2:ProxyRestriction

Functions

lasso_saml2_proxy_restriction_new ()

```
LassoNode~*
lasso_saml2_proxy_restriction_new (void);
```

Creates a new LassoSaml2ProxyRestriction object.

Returns

a newly created LassoSaml2ProxyRestriction object

Types and Values

struct LassoSaml2ProxyRestriction

```
struct LassoSaml2ProxyRestriction {
  LassoSaml2ConditionAbstract parent;

  /* elements */
  char *Audience;
  /* attributes */
  char *Count;
};
```

7.21 LassoSaml2StatementAbstract

LassoSaml2StatementAbstract — <saml2:StatementAbstract>

Functions

LassoNode *	lasso_saml2_statement_abstract_new ()

Types and Values

struct	LassoSaml2StatementAbstract

Description

```
<complexType name="StatementAbstractType" abstract="true"/>
<element name="AuthnStatement" type="saml:AuthnStatementType"/>
<complexType name="AuthnStatementType">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <sequence>
        <element ref="saml:SubjectLocality" minOccurs="0"/>
        <element ref="saml:AuthnContext"/>
      </sequence>
      <attribute name="AuthnInstant" type="dateTime" use="required"/>
      <attribute name="SessionIndex" type="string" use="optional"/>
      <attribute name="SessionNotOnOrAfter" type="dateTime" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.20: Schema fragment for saml2:StatementAbstract

Functions

lasso_saml2_statement_abstract_new ()

```
LassoNode~*
lasso_saml2_statement_abstract_new (void);
```

Creates a new LassoSaml2StatementAbstract object.

Returns

a newly created LassoSaml2StatementAbstract object

Types and Values

struct LassoSaml2StatementAbstract

```
struct LassoSaml2StatementAbstract {
  LassoNode parent;
};
```

7.22 LassoSaml2SubjectConfirmationData

LassoSaml2SubjectConfirmationData — <saml2:SubjectConfirmationData>

Functions

LassoNode *	lasso_saml2_subject_confirmation_data_new ()

Types and Values

struct	LassoSaml2SubjectConfirmationData

Description

```
<complexType name="SubjectConfirmationDataType" mixed="true">
  <complexContent>
    <restriction base="anyType">
      <sequence>
        <any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="NotBefore" type="dateTime" use="optional"/>
      <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
      <attribute name="Recipient" type="anyURI" use="optional"/>
      <attribute name="InResponseTo" type="NCName" use="optional"/>
      <attribute name="Address" type="string" use="optional"/>
      <anyAttribute namespace="##other" processContents="lax"/>
    </restriction>
  </complexContent>
</complexType>
```

Figure 7.21: Schema fragment for saml2:SubjectConfirmationData

Functions

lasso_saml2_subject_confirmation_data_new ()

```
LassoNode~*
lasso_saml2_subject_confirmation_data_new
                                (void);
```

Creates a new LassoSaml2SubjectConfirmationData object.

Returns

a newly created LassoSaml2SubjectConfirmationData object

Types and Values

struct LassoSaml2SubjectConfirmationData

```
struct LassoSaml2SubjectConfirmationData {
  LassoNode parent;

  /* attributes */
  char *NotBefore;
  char *NotOnOrAfter;
  char *Recipient;
  char *InResponseTo;
  char *Address;
};
```

7.23 LassoSaml2SubjectConfirmation

LassoSaml2SubjectConfirmation — <saml2:SubjectConfirmation>

Functions

LassoNode *	lasso_saml2_subject_confirmation_new ()

Types and Values

struct	LassoSaml2SubjectConfirmation

Description

```
<complexType name="SubjectConfirmationType">
  <sequence>
    <choice minOccurs="0">
      <element ref="saml:BaseID"/>
      <element ref="saml:NameID"/>
      <element ref="saml:EncryptedID"/>
    </choice>
    <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
  </sequence>
  <attribute name="Method" type="anyURI" use="required"/>
</complexType>
```

Figure 7.22: Schema fragment for saml2:SubjectConfirmation

Functions

lasso_saml2_subject_confirmation_new ()

```
LassoNode~*
lasso_saml2_subject_confirmation_new (void);
```

Creates a new LassoSaml2SubjectConfirmation object.

Returns

a newly created LassoSaml2SubjectConfirmation object

Types and Values

struct LassoSaml2SubjectConfirmation

```
struct LassoSaml2SubjectConfirmation {
  LassoNode parent;

  /* elements */
  LassoSaml2BaseIDAbstract *BaseID;
  LassoSaml2NameID *NameID;
  LassoSaml2EncryptedElement *EncryptedID;
  LassoSaml2SubjectConfirmationData *SubjectConfirmationData;
  /* attributes */
  char *Method;
};
```

7.24 LassoSaml2SubjectLocality

LassoSaml2SubjectLocality — <saml2:SubjectLocality>

Functions

LassoNode *	lasso_saml2_subject_locality_new ()

Types and Values

struct	LassoSaml2SubjectLocality

Description

```

<complexType name="SubjectLocalityType">
  <attribute name="Address" type="string" use="optional"/>
  <attribute name="DNSName" type="string" use="optional"/>
</complexType>
```

Figure 7.23: Schema fragment for saml2:SubjectLocality

Functions

lasso_saml2_subject_locality_new ()

```
LassoNode~*
lasso_saml2_subject_locality_new (void);
```

Creates a new LassoSaml2SubjectLocality object.

Returns

a newly created LassoSaml2SubjectLocality object

Types and Values

struct LassoSaml2SubjectLocality

```
struct LassoSaml2SubjectLocality {
  LassoNode parent;

  /* attributes */
  char *Address;
  char *DNSName;
};
```

7.25 LassoSaml2Subject

LassoSaml2Subject — <saml2:Subject>

Functions

LassoNode *	lasso_saml2_subject_new ()

Types and Values

struct	LassoSaml2Subject

Description

```
<complexType name="SubjectType">
  <choice>
    <sequence>
      <choice>
        <element ref="saml:BaseID"/>
        <element ref="saml:NameID"/>
        <element ref="saml:EncryptedID"/>
      </choice>
      <element ref="saml:SubjectConfirmation" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
  </choice>
</complexType>
```

Figure 7.24: Schema fragment for saml2:Subject

Functions

lasso_saml2_subject_new ()

```
LassoNode~*
lasso_saml2_subject_new (void);
```

Creates a new LassoSaml2Subject object.

Returns

a newly created LassoSaml2Subject object

Types and Values

struct LassoSaml2Subject

```
struct LassoSaml2Subject {
  LassoNode parent;

  /* elements */
  LassoSaml2BaseIDAbstract *BaseID;
  LassoSaml2NameID *NameID;
  LassoSaml2EncryptedElement *EncryptedID;
  LassoSaml2SubjectConfirmation *SubjectConfirmation;
};
```

7.26 LassoSamlp2ArtifactResolve

LassoSamlp2ArtifactResolve — <samlp2:ArtifactResolve>

Functions

LassoNode *	lasso_samlp2_artifact_resolve_new ()

Types and Values

struct	LassoSamlp2ArtifactResolve

Description

```
<complexType name="ArtifactResolveType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <element ref="samlp:Artifact"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.25: Schema fragment for samlp2:ArtifactResolve

Functions

lasso_samlp2_artifact_resolve_new ()

```
LassoNode~*
lasso_samlp2_artifact_resolve_new (void);
```

Creates a new LassoSamlp2ArtifactResolve object.

Returns

a newly created LassoSamlp2ArtifactResolve object

Types and Values

struct LassoSamlp2ArtifactResolve

```
struct LassoSamlp2ArtifactResolve {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  char *Artifact;
};
```

7.27 LassoSamlp2ArtifactResponse

LassoSamlp2ArtifactResponse — <samlp2:ArtifactResponse>

Functions

LassoNode *	lasso_samlp2_artifact_response_new ()

Types and Values

struct	LassoSamlp2ArtifactResponse

Description

```
<complexType name="ArtifactResponseType">
  <complexContent>
    <extension base="samlp:StatusResponseType">
      <sequence>
        <any namespace="##any" processContents="lax" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.26: Schema fragment for samlp2:ArtifactResponse

Functions

lasso_samlp2_artifact_response_new ()

```
LassoNode~*
lasso_samlp2_artifact_response_new (void);
```

Creates a new LassoSamlp2ArtifactResponse object.

Returns

a newly created LassoSamlp2ArtifactResponse object

Types and Values

struct LassoSamlp2ArtifactResponse

```
struct LassoSamlp2ArtifactResponse {
  LassoSamlp2StatusResponse parent;

  /* elements */
  LassoNode *any;
};
```

7.28 LassoSamlp2AssertionIDRequest

LassoSamlp2AssertionIDRequest —

Functions

LassoNode *	lasso_samlp2_assertion_id_request_new ()

Types and Values

struct	LassoSamlp2AssertionIDRequest

Description

Functions

lasso_samlp2_assertion_id_request_new ()

```
LassoNode~*
lasso_samlp2_assertion_id_request_new (void);
```

Creates a new LassoSamlp2AssertionIDRequest object.

Returns

a newly created LassoSamlp2AssertionIDRequest object

Types and Values

struct LassoSamlp2AssertionIDRequest

```
struct LassoSamlp2AssertionIDRequest {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  char *AssertionIDRef;
};
```

7.29 LassoSamlp2AttributeQuery

LassoSamlp2AttributeQuery — <samlp2:AttributeQuery>

Functions

LassoNode *	lasso_samlp2_attribute_query_new ()

Types and Values

struct	LassoSamlp2AttributeQuery

Description

```
<complexType name="AttributeQueryType">
  <complexContent>
    <extension base="samlp:SubjectQueryAbstractType">
      <sequence>
        <element ref="saml:Attribute" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.27: Schema fragment for samlp2:AttributeQuery

Functions

lasso_samlp2_attribute_query_new ()

```
LassoNode~*
lasso_samlp2_attribute_query_new (void);
```

Creates a new LassoSamlp2AttributeQuery object.

Returns

a newly created LassoSamlp2AttributeQuery object

Types and Values

struct LassoSamlp2AttributeQuery

```
struct LassoSamlp2AttributeQuery {
  LassoSamlp2SubjectQueryAbstract parent;

  /* elements */
  GList *Attribute; /* of LassoSaml2Attribute */
};
```

7.30 LassoSamlp2AuthnQuery

LassoSamlp2AuthnQuery — <samlp2:AuthnQuery>

Functions

LassoNode *	lasso_samlp2_authn_query_new ()

Types and Values

struct	LassoSamlp2AuthnQuery

Description

```
<complexType name="AuthnQueryType">
  <complexContent>
    <extension base="samlp:SubjectQueryAbstractType">
      <sequence>
        <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
      </sequence>
      <attribute name="SessionIndex" type="string" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.28: Schema fragment for samlp2:AuthnQuery

Functions

lasso_samlp2_authn_query_new ()

```
LassoNode~*
lasso_samlp2_authn_query_new (void);
```

Creates a new LassoSamlp2AuthnQuery object.

Returns

a newly created LassoSamlp2AuthnQuery object

Types and Values

struct LassoSamlp2AuthnQuery

```
struct LassoSamlp2AuthnQuery {
  LassoSamlp2SubjectQueryAbstract parent;

  /* elements */
  LassoSamlp2RequestedAuthnContext *RequestedAuthnContext;
  /* attributes */
  char *SessionIndex;
};
```

7.31 LassoSamlp2AuthnRequest

LassoSamlp2AuthnRequest — <samlp2:AuthnRequest>

Functions

LassoNode *	lasso_samlp2_authn_request_new ()

Types and Values

struct	LassoSamlp2AuthnRequest

Description

```
<complexType name="AuthnRequestType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <element ref="saml:Subject" minOccurs="0"/>
        <element ref="samlp:NameIDPolicy" minOccurs="0"/>
        <element ref="saml:Conditions" minOccurs="0"/>
        <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
        <element ref="samlp:Scoping" minOccurs="0"/>
      </sequence>
      <attribute name="ForceAuthn" type="boolean" use="optional"/>
      <attribute name="IsPassive" type="boolean" use="optional"/>
      <attribute name="ProtocolBinding" type="anyURI" use="optional"/>
      <attribute name="AssertionConsumerServiceIndex" type="unsignedShort" use="optional"/>
      <attribute name="AssertionConsumerServiceURL" type="anyURI" use="optional"/>
      <attribute name="AttributeConsumingServiceIndex" type="unsignedShort" use="optional ↩
          "/>
      <attribute name="ProviderName" type="string" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.29: Schema fragment for samlp2:AuthnRequest

Functions

lasso_samlp2_authn_request_new ()

```
LassoNode~*
lasso_samlp2_authn_request_new (void);
```

Creates a new LassoSamlp2AuthnRequest object.

Returns

a newly created LassoSamlp2AuthnRequest object

Types and Values

struct LassoSamlp2AuthnRequest

```
struct LassoSamlp2AuthnRequest {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  LassoSaml2Subject *Subject;
  LassoSamlp2NameIDPolicy *NameIDPolicy;
  LassoSaml2Conditions *Conditions;
  LassoSamlp2RequestedAuthnContext *RequestedAuthnContext;
  LassoSamlp2Scoping *Scoping;
  /* attributes */
  gboolean ForceAuthn;
  gboolean IsPassive;
  char *ProtocolBinding;
  int AssertionConsumerServiceIndex;
  char *AssertionConsumerServiceURL;
  int AttributeConsumingServiceIndex;
  char *ProviderName;

  /* This field is deprecated do not use it,
   * kept for ABI compatibility */
};
```

7.32 LassoSamlp2AuthzDecisionQuery

LassoSamlp2AuthzDecisionQuery — <samlp2:AuthzDecisionQuery>

Functions

LassoNode *	lasso_samlp2_authz_decision_query_new ()

Types and Values

struct	LassoSamlp2AuthzDecisionQuery

Description

```
<complexType name="AuthzDecisionQueryType">
  <complexContent>
    <extension base="samlp:SubjectQueryAbstractType">
      <sequence>
        <element ref="saml:Action" maxOccurs="unbounded"/>
        <element ref="saml:Evidence" minOccurs="0"/>
      </sequence>
      <attribute name="Resource" type="anyURI" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.30: Schema fragment for samlp2:AuthzDecisionQuery

Functions

lasso_samlp2_authz_decision_query_new ()

```
LassoNode~*
lasso_samlp2_authz_decision_query_new (void);
```

Creates a new LassoSamlp2AuthzDecisionQuery object.

Returns

a newly created LassoSamlp2AuthzDecisionQuery object

Types and Values

struct LassoSamlp2AuthzDecisionQuery

```
struct LassoSamlp2AuthzDecisionQuery {
  LassoSamlp2SubjectQueryAbstract parent;

  /* elements */
  LassoSaml2Action *Action;
  LassoSaml2Evidence *Evidence;
  /* attributes */
  char *Resource;
};
```

7.33 LassoSamlp2Extensions

LassoSamlp2Extensions — <samlp2:Extensions>

Functions

LassoNode *	lasso_samlp2_extensions_new ()

Types and Values

struct	LassoSamlp2Extensions

Description

```
<complexType name="ExtensionsType">
  <sequence>
    <any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Figure 7.31: Schema fragment for samlp2:Extensions

To specify the content of this object you must attach an xmlNode to it using lasso_node_set_original_xmlnode(). The content (attributes, children and namespaces) of this node will be copied to the result node returned by calls to lasso_node_get_xmlNode().

Functions

lasso_samlp2_extensions_new ()

```
LassoNode~*
lasso_samlp2_extensions_new (void);
```

Creates a new LassoSamlp2Extensions object.

Returns

a newly created LassoSamlp2Extensions object

Types and Values

struct LassoSamlp2Extensions

```
struct LassoSamlp2Extensions {
  LassoNode parent;
};
```

7.34 LassoSamlp2IDPEntry

LassoSamlp2IDPEntry — <samlp2:IDPEntry>

Functions

LassoNode *	lasso_samlp2_idp_entry_new ()

Types and Values

struct	LassoSamlp2IDPEntry

Description

```
<complexType name="IDPEntryType">
  <attribute name="ProviderID" type="anyURI" use="required"/>
  <attribute name="Name" type="string" use="optional"/>
  <attribute name="Loc" type="anyURI" use="optional"/>
</complexType>
```

Figure 7.32: Schema fragment for samlp2:IDPEntry

Functions

lasso_samlp2_idp_entry_new ()

```
LassoNode~*
lasso_samlp2_idp_entry_new (void);
```

Creates a new LassoSamlp2IDPEntry object.

Returns

a newly created LassoSamlp2IDPEntry object

Types and Values

struct LassoSamlp2IDPEntry

```
struct LassoSamlp2IDPEntry {
  LassoNode parent;

  /* attributes */
  char *ProviderID;
  char *Name;
  char *Loc;
};
```

7.35 LassoSamlp2IDPList

LassoSamlp2IDPList — <samlp2:IDPList>

Functions

LassoNode *	lasso_samlp2_idp_list_new ()

Types and Values

struct	LassoSamlp2IDPList

Description

```
<complexType name="IDPListType">
  <sequence>
    <element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
    <element ref="samlp:GetComplete" minOccurs="0"/>
  </sequence>
</complexType>
```

Figure 7.33: Schema fragment for samlp2:IDPList

Functions

lasso_samlp2_idp_list_new ()

```
LassoNode~*
lasso_samlp2_idp_list_new (void);
```

Creates a new LassoSamlp2IDPList object.

Returns

a newly created LassoSamlp2IDPList object

Types and Values

struct LassoSamlp2IDPList

```
struct LassoSamlp2IDPList {
  LassoNode parent;

  /* elements */
  GList *IDPEntry; /* of LassoSamlp2IDPEntry */
  char *GetComplete;
};
```

7.36 LassoSamlp2LogoutRequest

LassoSamlp2LogoutRequest — <samlp2:LogoutRequest>

Functions

LassoNode *	lasso_samlp2_logout_request_new ()
GList *	lasso_samlp2_logout_request_get_session_indexes ()
void	lasso_samlp2_logout_request_set_session_indexes ()

Types and Values

struct	LassoSamlp2LogoutRequest

Description

```
<complexType name="LogoutRequestType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <choice>
          <element ref="saml:BaseID"/>
          <element ref="saml:NameID"/>
          <element ref="saml:EncryptedID"/>
        </choice>
        <element ref="samlp:SessionIndex" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="Reason" type="string" use="optional"/>
      <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.34: Schema fragment for samlp2:LogoutRequest

Functions

lasso_samlp2_logout_request_new ()

```
LassoNode~*
lasso_samlp2_logout_request_new (void);
```

Creates a new LassoSamlp2LogoutRequest object.

Returns

a newly created LassoSamlp2LogoutRequest object

lasso_samlp2_logout_request_get_session_indexes ()

```
GList~*
lasso_samlp2_logout_request_get_session_indexes
                               (LassoSamlp2LogoutRequest *logout_request);
```

If the logout request contains more than one SessionIndex element, this method must be used to retrieve due to historical circonstances. It will a return a list of the content of the SessionIndex elements.

Parameters

logout_request	a LogoutRequest object	

Returns

a GList of sessions index.

[element-type utf8][transfer full]

lasso_samlp2_logout_request_set_session_indexes ()

```
void
lasso_samlp2_logout_request_set_session_indexes
                               (LassoSamlp2LogoutRequest *logout_request,
                                GList *session_index);
```

If you want to set more than one SessionIndex on a LogoutRequest, use this method. Beware that the public field named SessionIndex corresponds to the last element in this list. This is an symptom of the way elements are parsed by Lasso.

Parameters

logout_request	a LogoutRequest object	
session_index	a list of session index.	*[element-type utf8]*

Types and Values

struct LassoSamlp2LogoutRequest

```
struct LassoSamlp2LogoutRequest {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  LassoSaml2BaseIDAbstract *BaseID;
  LassoSaml2NameID *NameID;
  LassoSaml2EncryptedElement *EncryptedID;
  char *SessionIndex;
  /* attributes */
  char *Reason;
  char *NotOnOrAfter;

  /* This field is deprecated do not use it,
   * kept for ABI compatibility */
};
```

7.37 LassoSamlp2LogoutResponse

LassoSamlp2LogoutResponse — <samlp2:LogoutResponse>

Functions

LassoNode *	lasso_samlp2_logout_response_new ()

Types and Values

struct	LassoSamlp2LogoutResponse

Description

```
<element name="LogoutResponse" type="samlp:StatusResponseType"/>
```

Figure 7.35: Schema fragment for samlp2:LogoutResponse

Functions

lasso_samlp2_logout_response_new ()

```
LassoNode~*
lasso_samlp2_logout_response_new (void);
```

Creates a new LassoSamlp2LogoutResponse object.

Returns

a newly created LassoSamlp2LogoutResponse object

Types and Values

struct LassoSamlp2LogoutResponse

```
struct LassoSamlp2LogoutResponse {
  LassoSamlp2StatusResponse parent;

  /* This field is deprecated do not use it,
   * kept for ABI compatibility */
};
```

7.38 LassoSamlp2ManageNameIDRequest

LassoSamlp2ManageNameIDRequest — <samlp2:ManageNameIDRequest>

Functions

LassoNode *	lasso_samlp2_manage_name_id_request_new ()

Types and Values

struct	LassoSamlp2ManageNameIDRequest

Description

```
<complexType name="ManageNameIDRequestType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <choice>
          <element ref="saml:NameID"/>
          <element ref="saml:EncryptedID"/>
        </choice>
        <choice>
          <element ref="samlp:NewID"/>
          <element ref="samlp:NewEncryptedID"/>
          <element ref="samlp:Terminate"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.36: Schema fragment for samlp2:ManageNameIDRequest

Functions

lasso_samlp2_manage_name_id_request_new ()

```
LassoNode~*
lasso_samlp2_manage_name_id_request_new
                                (void);
```

Creates a new LassoSamlp2ManageNameIDRequest object.

Returns

a newly created LassoSamlp2ManageNameIDRequest object

Types and Values

struct LassoSamlp2ManageNameIDRequest

```
struct LassoSamlp2ManageNameIDRequest {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  LassoSaml2NameID *NameID;
  LassoSaml2EncryptedElement *EncryptedID;
  char *NewID;
  LassoSaml2EncryptedElement *NewEncryptedID;
  LassoSamlp2Terminate *Terminate;
};
```

7.39 LassoSamlp2ManageNameIDResponse

LassoSamlp2ManageNameIDResponse — <samlp2:ManageNameIDResponse>

Functions

LassoNode *	lasso_samlp2_manage_name_id_response_new ()

Types and Values

struct	LassoSamlp2ManageNameIDResponse

Description

```
<element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
```

Figure 7.37: Schema fragment for samlp2:ManageNameIDResponse

Functions

lasso_samlp2_manage_name_id_response_new ()

```
LassoNode~*
lasso_samlp2_manage_name_id_response_new
                                (void);
```

Creates a new LassoSamlp2ManageNameIDResponse object.

Returns

a newly created LassoSamlp2ManageNameIDResponse object

Types and Values

struct LassoSamlp2ManageNameIDResponse

```
struct LassoSamlp2ManageNameIDResponse {
  LassoSamlp2StatusResponse parent;
};
```

7.40 LassoSamlp2NameIDMappingRequest

LassoSamlp2NameIDMappingRequest — <samlp2:NameIDMappingRequest>

Functions

LassoNode *	lasso_samlp2_name_id_mapping_request_new ()

Types and Values

struct	LassoSamlp2NameIDMappingRequest

Description

```
<complexType name="NameIDMappingRequestType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <choice>
          <element ref="saml:BaseID"/>
          <element ref="saml:NameID"/>
          <element ref="saml:EncryptedID"/>
        </choice>
        <element ref="samlp:NameIDPolicy"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.38: Schema fragment for samlp2:NameIDMappingRequest

Functions

lasso_samlp2_name_id_mapping_request_new ()

```
LassoNode~*
lasso_samlp2_name_id_mapping_request_new
                                (void);
```

Creates a new LassoSamlp2NameIDMappingRequest object.

Returns

a newly created LassoSamlp2NameIDMappingRequest object

Types and Values

struct LassoSamlp2NameIDMappingRequest

```
struct LassoSamlp2NameIDMappingRequest {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  LassoSaml2BaseIDAbstract *BaseID;
  LassoSaml2NameID *NameID;
  LassoSaml2EncryptedElement *EncryptedID;
  LassoSamlp2NameIDPolicy *NameIDPolicy;
};
```

7.41 LassoSamlp2NameIDMappingResponse

LassoSamlp2NameIDMappingResponse — <samlp2:NameIDMappingResponse>

Functions

LassoNode *	lasso_samlp2_name_id_mapping_response_new ()

Types and Values

struct	LassoSamlp2NameIDMappingResponse

Description

```
<complexType name="NameIDMappingResponseType">
  <complexContent>
    <extension base="samlp:StatusResponseType">
      <choice>
        <element ref="saml:NameID"/>
        <element ref="saml:EncryptedID"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.39: Schema fragment for samlp2:NameIDMappingResponse

Functions

lasso_samlp2_name_id_mapping_response_new ()

```
LassoNode~*
lasso_samlp2_name_id_mapping_response_new
                               (void);
```

Creates a new LassoSamlp2NameIDMappingResponse object.

Returns

a newly created LassoSamlp2NameIDMappingResponse object

Types and Values

struct LassoSamlp2NameIDMappingResponse

```
struct LassoSamlp2NameIDMappingResponse {
  LassoSamlp2StatusResponse parent;
  /* elements */
  LassoSaml2NameID *NameID;
  LassoSaml2EncryptedElement *EncryptedID;
};
```

7.42 LassoSamlp2NameIDPolicy

LassoSamlp2NameIDPolicy — <samlp2:NameIDPolicy>

Functions

LassoNode *	lasso_samlp2_name_id_policy_new ()

Types and Values

struct	LassoSamlp2NameIDPolicy

Description

```
<complexType name="NameIDPolicyType">
  <attribute name="Format" type="anyURI" use="optional"/>
  <attribute name="SPNameQualifier" type="string" use="optional"/>
  <attribute name="AllowCreate" type="boolean" use="optional"/>
</complexType>
```

Figure 7.40: Schema fragment for samlp2:NameIDPolicy

Functions

lasso_samlp2_name_id_policy_new ()

```
LassoNode~*
lasso_samlp2_name_id_policy_new (void);
```

Creates a new LassoSamlp2NameIDPolicy object.

Returns

a newly created LassoSamlp2NameIDPolicy object

Types and Values

struct LassoSamlp2NameIDPolicy

```
struct LassoSamlp2NameIDPolicy {
  LassoNode parent;

  /* attributes */
  char *Format;
  char *SPNameQualifier;
  gboolean AllowCreate;
};
```

NameIDPolicy element is part of AuthnRequest and allows to give hint to the identity provider about the NameID the service provider desire to receive.

Members

LassoNode *parent*;		
char **Format*;		the name-identifier format, see LASSO_SAML2_NAME_IDENTIFIER_FORMAT_E LASSO_SAML2_NAME_IDENTIFIER_FORMAT_X LASSO_SAML2_NAME_IDENTIFIER_FORMAT_W LASSO_SAML2_NAME_IDENTIFIER_FORMAT_K LASSO_SAML2_NAME_IDENTIFIER_FORMAT_E LASSO_SAML2_NAME_IDENTIFIER_FORMAT_P LASSO_SAML2_NAME_IDENTIFIER_FORMAT_T or LASSO_SAML2_NAME_IDENTIFIER_FORMAT_E

char **SPNameQualifier*;	other providerID to use as SPNameQualifier in the returned assertion. By default an IdP create a NameID for the requesting entity, this field allows to obtain a NameID qualified for a third party.

gboolean *AllowCreate*;	authorize the identity provider to create a new federation, otherwise it must report a failure in case no existing federation is found (status code of LASSO_LIB_STATUS_CODE_FEDERATION_DOES and error code of LASSO_PROFILE_ERROR_FEDERATION_NOT_FO

7.43 LassoSamlp2RequestAbstract

LassoSamlp2RequestAbstract — <samlp2:RequestAbstract>

Functions

LassoNode *	lasso_samlp2_request_abstract_new ()

Types and Values

struct	LassoSamlp2RequestAbstract

Description

```
<complexType name="RequestAbstractType" abstract="true">
  <sequence>
    <element ref="saml:Issuer" minOccurs="0"/>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="samlp:Extensions" minOccurs="0"/>
  </sequence>
  <attribute name="ID" type="ID" use="required"/>
  <attribute name="Version" type="string" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
  <attribute name="Destination" type="anyURI" use="optional"/>
  <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>
```

Figure 7.41: Schema fragment for samlp2:RequestAbstract

Functions

lasso_samlp2_request_abstract_new ()

```
LassoNode~*
lasso_samlp2_request_abstract_new (void);
```

Creates a new LassoSamlp2RequestAbstract object.

Returns

a newly created LassoSamlp2RequestAbstract object

Types and Values

struct LassoSamlp2RequestAbstract

```
struct LassoSamlp2RequestAbstract {
  LassoNode parent;

  /* elements */
  LassoSaml2NameID *Issuer;
  LassoSamlp2Extensions *Extensions;
  /* attributes */
  char *ID;
  char *Version;
  char *IssueInstant;
  char *Destination;
  char *Consent;
};
```

7.44 LassoSamlp2RequestedAuthnContext

LassoSamlp2RequestedAuthnContext — <samlp2:RequestedAuthnContext>

Functions

LassoNode *	lasso_samlp2_requested_authn_context_new ()

Types and Values

struct	LassoSamlp2RequestedAuthnContext

Description

```
<complexType name="RequestedAuthnContextType">
  <choice>
    <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
    <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
  </choice>
  <attribute name="Comparison" type="samlp:AuthnContextComparisonType" use="optional"/>
</complexType>
```

Figure 7.42: Schema fragment for samlp2:RequestedAuthnContext

Functions

lasso_samlp2_requested_authn_context_new ()

```
LassoNode~*
lasso_samlp2_requested_authn_context_new
                               (void);
```

Creates a new LassoSamlp2RequestedAuthnContext object.

Returns

a newly created LassoSamlp2RequestedAuthnContext object

Types and Values

struct LassoSamlp2RequestedAuthnContext

```
struct LassoSamlp2RequestedAuthnContext {
  LassoNode parent;

  /* elements */
  GList *AuthnContextClassRef; /* of strings */
  GList *AuthnContextDeclRef; /* of strings */
  /* attributes */
  char *Comparison;
};
```

7.45 LassoSamlp2Response

LassoSamlp2Response — <samlp2:Response>

Functions

LassoNode *	lasso_samlp2_response_new ()

Types and Values

struct	LassoSamlp2Response

Description

```

<complexType name="ResponseType">
  <complexContent>
    <extension base="samlp:StatusResponseType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="saml:Assertion"/>
        <element ref="saml:EncryptedAssertion"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.43: Schema fragment for samlp2:Response

Functions

lasso_samlp2_response_new ()

```
LassoNode~*
lasso_samlp2_response_new (void);
```

Creates a new LassoSamlp2Response object.

Returns

a newly created LassoSamlp2Response object

Types and Values

struct LassoSamlp2Response

```
struct LassoSamlp2Response {
  LassoSamlp2StatusResponse parent;

  /* elements */
  GList *Assertion; /* of LassoSaml2Assertion */
  GList *EncryptedAssertion; /* of LassoSaml2EncryptedElement */
};
```

7.46 LassoSamlp2Scoping

LassoSamlp2Scoping — <samlp2:Scoping>

Functions

LassoNode *	lasso_samlp2_scoping_new ()

Types and Values

struct	LassoSamlp2Scoping

Description

```

<complexType name="ScopingType">
  <sequence>
    <element ref="samlp:IDPList" minOccurs="0"/>
    <element ref="samlp:RequesterID" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="ProxyCount" type="nonNegativeInteger" use="optional"/>
</complexType>
```

Figure 7.44: Schema fragment for samlp2:Scoping

Functions

lasso_samlp2_scoping_new ()

```
LassoNode~*
lasso_samlp2_scoping_new (void);
```

Creates a new LassoSamlp2Scoping object.

Returns

a newly created LassoSamlp2Scoping object

Types and Values

struct LassoSamlp2Scoping

```
struct LassoSamlp2Scoping {
  LassoNode parent;

  /* elements */
  LassoSamlp2IDPList *IDPList;
  char *RequesterID;
  /* attributes */
  char *ProxyCount;
};
```

7.47 LassoSamlp2StatusCode

LassoSamlp2StatusCode — <samlp2:StatusCode>

Functions

LassoNode *	lasso_samlp2_status_code_new ()

Types and Values

struct	LassoSamlp2StatusCode

Description

```

<complexType name="StatusCodeType">
  <sequence>
    <element ref="samlp:StatusCode" minOccurs="0"/>
  </sequence>
  <attribute name="Value" type="anyURI" use="required"/>
</complexType>
```

Figure 7.45: Schema fragment for samlp2:StatusCode

Functions

lasso_samlp2_status_code_new ()

```
LassoNode~*
lasso_samlp2_status_code_new (void);
```

Creates a new LassoSamlp2StatusCode object.

Returns

a newly created LassoSamlp2StatusCode object

Types and Values

struct LassoSamlp2StatusCode

```
struct LassoSamlp2StatusCode {
  LassoNode parent;

  /* elements */
  LassoSamlp2StatusCode *StatusCode;
  /* attributes */
  char *Value;
};
```

7.48 LassoSamlp2StatusDetail

LassoSamlp2StatusDetail — <samlp2:StatusDetail>

Functions

LassoNode *	lasso_samlp2_status_detail_new ()

Types and Values

struct	LassoSamlp2StatusDetail

Description

```

<complexType name="StatusDetailType">
  <sequence>
    <any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Figure 7.46: Schema fragment for samlp2:StatusDetail

Functions

lasso_samlp2_status_detail_new ()

```
LassoNode~*
lasso_samlp2_status_detail_new (void);
```

Creates a new LassoSamlp2StatusDetail object.

Returns

a newly created LassoSamlp2StatusDetail object

Types and Values

struct LassoSamlp2StatusDetail

```
struct LassoSamlp2StatusDetail {
  LassoNode parent;
};
```

7.49 LassoSamlp2StatusResponse

LassoSamlp2StatusResponse — <samlp2:StatusResponse>

Functions

LassoNode *	lasso_samlp2_status_response_new ()

Types and Values

struct	LassoSamlp2StatusResponse

Description

```
<complexType name="StatusResponseType">
  <sequence>
    <element ref="saml:Issuer" minOccurs="0"/>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="samlp:Extensions" minOccurs="0"/>
    <element ref="samlp:Status"/>
  </sequence>
  <attribute name="ID" type="ID" use="required"/>
  <attribute name="InResponseTo" type="NCName" use="optional"/>
  <attribute name="Version" type="string" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
  <attribute name="Destination" type="anyURI" use="optional"/>
  <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>
```

Figure 7.47: Schema fragment for samlp2:StatusResponse

Functions

lasso_samlp2_status_response_new ()

```
LassoNode~*
lasso_samlp2_status_response_new (void);
```

Creates a new LassoSamlp2StatusResponse object.

Returns

a newly created LassoSamlp2StatusResponse object

Types and Values

struct LassoSamlp2StatusResponse

```
struct LassoSamlp2StatusResponse {
  LassoNode parent;

  /* elements */
  LassoSaml2NameID *Issuer;
  LassoSamlp2Extensions *Extensions;
  LassoSamlp2Status *Status;
  /* attributes */
  char *ID;
  char *InResponseTo;
  char *Version;
  char *IssueInstant;
  char *Destination;
  char *Consent;
};
```

7.50 LassoSamlp2Status

LassoSamlp2Status — <samlp2:Status>

Functions

LassoNode *	lasso_samlp2_status_new ()

Types and Values

struct	LassoSamlp2Status

Description

```
<complexType name="StatusType">
  <sequence>
    <element ref="samlp:StatusCode"/>
    <element ref="samlp:StatusMessage" minOccurs="0"/>
    <element ref="samlp:StatusDetail" minOccurs="0"/>
  </sequence>
</complexType>
```

Figure 7.48: Schema fragment for samlp2:Status

Functions

lasso_samlp2_status_new ()

```
LassoNode~*
lasso_samlp2_status_new (void);
```

Creates a new LassoSamlp2Status object.

Returns

a newly created LassoSamlp2Status object

Types and Values

struct LassoSamlp2Status

```
struct LassoSamlp2Status {
  LassoNode parent;

  /* elements */
  LassoSamlp2StatusCode *StatusCode;
  char *StatusMessage;
  LassoSamlp2StatusDetail *StatusDetail;
};
```

7.51 LassoSamlp2SubjectQueryAbstract

LassoSamlp2SubjectQueryAbstract — <samlp2:SubjectQueryAbstract>

Functions

LassoNode *	lasso_samlp2_subject_query_abstract_new ()

Types and Values

struct	LassoSamlp2SubjectQueryAbstract

Description

```
<complexType name="SubjectQueryAbstractType" abstract="true">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <element ref="saml:Subject"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 7.49: Schema fragment for samlp2:SubjectQueryAbstract

Functions

lasso_samlp2_subject_query_abstract_new ()

```
LassoNode~*
lasso_samlp2_subject_query_abstract_new
                                (void);
```

Creates a new LassoSamlp2SubjectQueryAbstract object.

Returns

a newly created LassoSamlp2SubjectQueryAbstract object

Types and Values

struct LassoSamlp2SubjectQueryAbstract

```
struct LassoSamlp2SubjectQueryAbstract {
  LassoSamlp2RequestAbstract parent;

  /* elements */
  LassoSaml2Subject *Subject;
};
```

7.52 LassoSamlp2Terminate

LassoSamlp2Terminate — <samlp2:Terminate>

Functions

LassoNode *	lasso_samlp2_terminate_new ()

Types and Values

struct	LassoSamlp2Terminate

Description

```
<complexType name="TerminateType"/>
```

Figure 7.50: Schema fragment for samlp2:Terminate

Functions

lasso_samlp2_terminate_new ()

```
LassoNode~*
lasso_samlp2_terminate_new (void);
```

Creates a new LassoSamlp2Terminate object.

Returns

a newly created LassoSamlp2Terminate object

Types and Values

struct LassoSamlp2Terminate

```
struct LassoSamlp2Terminate {
  LassoNode parent;
};
```

7.53 LassoEcpRequest

LassoEcpRequest — <ecp:Request>

Functions

int	lasso_ecp_request_validate ()
LassoNode *	lasso_ecp_request_new ()

Description

```
<element name="Request" type="ecp:RequestType"/>
<complexType name="RequestType">
    <sequence>
        <element ref="saml:Issuer"/>
        <element ref="samlp:IDPList" minOccurs="0"/>
    </sequence>
    <attribute ref="S:mustUnderstand" use="required"/>
    <attribute ref="S:actor" use="required"/>
    <attribute name="ProviderName" type="string" use="optional"/>
    <attribute name="IsPassive" type="boolean" use="optional"/>
</complexType>
```

Figure 7.51: Schema fragment for ecp:Request

Functions

lasso_ecp_request_validate ()

```
int
lasso_ecp_request_validate (LassoEcpRequest *request);
```

Validates the LassoEcpRequest object conforms to required values.

- mustUnderstand must be TRUE
- actor must be equal to LASSO_SOAP_ENV_ACTOR

Parameters

request	the LassoEcpRequest object to validate

Returns

0 on success, error code otherwise

lasso_ecp_request_new ()

```
LassoNode~*
lasso_ecp_request_new (const gchar *Issuer,
                       gboolean IsPassive,
                       const gchar *ProviderName,
                       LassoSamlp2IDPList *IDPList);
```

Creates and intializes new LassoEcpRequest object.

The LassoEcpRequest object is initialized as follows:

```
Issuer = Issuer
IsPassive = IsPassive
```

```
ProviderName = ProviderName
IDPList = IDPList (if non-NULL)
mustUnderstand = TRUE
actor = LASSO_SOAP_ENV_ACTOR
```

Parameters

IDPList	.	*[allow-none]*

Returns

a newly created and initialized LassoEcpRequest object

7.54 LassoEcpResponse

LassoEcpResponse — <ecp:Response>

Functions

int	lasso_ecp_response_validate ()
LassoNode *	lasso_ecp_response_new ()

Description

```
<element name="Response" type="ecp:ResponseType"/>
<complexType name="ResponseType">
    <attribute ref="S:mustUnderstand" use="required"/>
    <attribute ref="S:actor" use="required"/>
    <attribute name="AssertionConsumerServiceURL" type="anyURI" use="required"/>
</complexType>
```

Figure 7.52: Schema fragment for ecp:Response

Functions

lasso_ecp_response_validate ()

```
int
lasso_ecp_response_validate (LassoEcpResponse *response);
```

Validates the LassoEcpResponse object conforms to required values.

- AssertionConsumerServiceURL must be non-NULL
- mustUnderstand must be TRUE
- actor must be equal to LASSO_SOAP_ENV_ACTOR

Parameters

response	The LassoEcpResponse object to validate

Returns

0 on success, error code otherwise

lasso_ecp_response_new ()

```
LassoNode~*
lasso_ecp_response_new (const gchar *AssertionConsumerServiceURL);
```

Creates and initializes a new LassoEcpResponse object.

The # object is initialized as follows:

```
AssertionConsumerServiceURL = AssertionConsumerServiceURL
mustUnderstand = TRUE
actor = LASSO_SOAP_ENV_ACTOR
```

Parameters

AssertionConsumerServiceURL	.	*[allow-none]*

Returns

a newly created and initialized LassoEcpResponse object

7.55 LassoEcpRelayState

LassoEcpRelayState — <ecp:RelayState>

Functions

int	lasso_ecp_relay_state_validate ()
LassoNode *	lasso_ecp_relay_state_new ()

Description

```
<element name="RelayState" type="ecp:RelayStateType"/>
<complexType name="RelayStateType">
    <simpleContent>
        <extension base="string">
            <attribute ref="S:mustUnderstand" use="required"/>
            <attribute ref="S:actor" use="required"/>
        </extension>
    </simpleContent>
</complexType>
```

Figure 7.53: Schema fragment for ecp:RelayState

Functions

lasso_ecp_relay_state_validate ()

```
int
lasso_ecp_relay_state_validate (LassoEcpRelayState *relaystate);
```

Validates the LassoEcpRelayState object conforms to required values.

- RelayState must be non-NULL
- mustUnderstand must be TRUE
- actor must be equal to LASSO_SOAP_ENV_ACTOR

Parameters

relaystate	The LassoEcpRelayState

Returns

0 on success, error code otherwise

lasso_ecp_relay_state_new ()

```
LassoNode~*
lasso_ecp_relay_state_new (const gchar *RelayState);
```

The LassoEcpRelayState object is initialized as follows:

```
RelayState = RelayState (if non-NULL)
mustUnderstand = TRUE
actor = LASSO_SOAP_ENV_ACTOR
```

Parameters

RelayState	.	*[allow-none]*

Returns

a newly created and initialized LassoEcpRelayState object

Chapter 8

Identity Web Services Framework 1.0

8.1 LassoDiscovery

LassoDiscovery — ID-WSF Discovery Service Profile

Functions

LassoDiscovery *	lasso_discovery_new ()
LassoDiscovery *	lasso_discovery_new_full ()
lasso_error_t	lasso_discovery_init_modify ()
lasso_error_t	lasso_discovery_add_insert_entry ()
lasso_error_t	lasso_discovery_add_remove_entry ()
lasso_error_t	lasso_discovery_init_query ()
lasso_error_t	lasso_discovery_add_requested_service_type ()
lasso_error_t	lasso_discovery_process_request_msg ()
lasso_error_t	lasso_discovery_build_response_msg ()
lasso_error_t	lasso_discovery_process_modify_response_msg ()
lasso_error_t	lasso_discovery_process_query_response_msg ()
LassoWsfProfile *	lasso_discovery_get_service ()
GList *	lasso_discovery_get_services ()
LassoWsfProfile *	(*LassoWsfProfileConstructor) ()
void	lasso_discovery_register_constructor_for_service_type ()
void	lasso_discovery_unregister_constructor_for_service_type ()

Types and Values

struct	LassoDiscovery

Description

The Discovery service usually runs on the principal identity provider and knowns about resources and services related to the principal. Attribute providers can register themselves as offering resources for an user while other services can ask where to find a given resource.

The following example is a service provider asking for a "PP" service (an attribute provider for the "Personal Profile"):

```
LassoServer *server;  // initialized before
char* session_dump;   // initialized before
```

```
LassoDiscovery *discovery;    // discovery service
char *soap_answer;            // SOAP answer from disco service
LassoProfileService *service; // instance to perform on requested service

discovery = lasso_discovery_new(server);
lasso_wsf_profile_set_session_from_dump(LASSO_WSF_PROFILE(discovery), session_dump);
lasso_discovery_init_query(discovery);
lasso_discovery_add_requested_service(discovery, LASSO_PP10_HREF);
lasso_discovery_build_request_msg(discovery);

// service must perform SOAP call to LASSO_WSF_PROFILE(discovery)->msg_url
// the SOAP message is LASSO_WSF_PROFILE(discovery)->msg_body.  The answer
// is stored in char* soap_answer;

lasso_discovery_process_query_response_msg(discovery, soap_answer);

// get an object to access the first service returned, or NULL if an error happened
service = lasso_discovery_get_service(discovery);
```

Functions

lasso_discovery_new ()

```
LassoDiscovery~*
lasso_discovery_new (LassoServer *server);
```

Creates a new LassoDiscovery.

Parameters

server	the LassoServer

Returns

a newly created LassoDiscovery object; or NULL if an error occured.

lasso_discovery_new_full ()

```
LassoDiscovery~*
lasso_discovery_new_full (LassoServer *server,
                          LassoDiscoResourceOffering *offering);
```

Creates a new LassoDiscovery.

Parameters

server	the LassoServer
offering	the LassoDiscoResourceOffering

Returns

a newly created LassoDiscovery object; or NULL if an error occured.

lasso_discovery_init_modify ()

```
lasso_error_t
lasso_discovery_init_modify (LassoDiscovery *discovery,
                             const char *security_mech_id);
```

Initializes a disco Modify/InsertEntry

Parameters

discovery	a LassoDiscovery	
security_mech_id	the security mechanism identifier.	*[allow-none]*

Returns

0 on success; an error code otherwise.

lasso_discovery_add_insert_entry ()

```
lasso_error_t
lasso_discovery_add_insert_entry (LassoDiscovery *discovery,
                                  LassoDiscoServiceInstance *serviceInstance,
                                  LassoDiscoResourceID *resourceId);
```

Add an LassoDiscoInsertEntry containing a new LassoDiscoResourceOffering, and initialize the LassoDiscoResourceOffering using the LassoDiscoServiceInstance and LassoDiscoResourceId object.

Parameters

discovery	a LassoDiscovery object	
serviceInstance	an optional LassoDiscoServiceInstance object	
resourceID	the new LassoDiscoResourceID used to create the LassoDiscoResrouceOffering	

Returns

the newly created LassoDiscoInsertEntry or NULL if some preconditions failed.

lasso_discovery_add_remove_entry ()

```
lasso_error_t
lasso_discovery_add_remove_entry (LassoDiscovery *discovery,
                                  const gchar *entryID);
```

Add a RemoveEntry to the current Modify message for a Discovery service, to remove the resource offering identified by entryID (returned in the response to a Modify/InsertEntry message).

Parameters

discovery	a LassoDiscovery object	
entryID	the idenitfier of a ResourceOffering to remove.	

Returns

0 if successful, an error code otherwise.

lasso_discovery_init_query ()

```
lasso_error_t
lasso_discovery_init_query (LassoDiscovery *discovery,
                            const gchar *security_mech_id);
```

Initializes a disco:Query message.

Parameters

discovery	a LassoDiscovery	
security_mech_id	identifier of a wished security mechanism, or NULL if any is ok.	*[allow-none]*

Returns

0 on success; or a negative value otherwise.

lasso_discovery_add_requested_service_type ()

```
lasso_error_t
lasso_discovery_add_requested_service_type
                               (LassoDiscovery *discovery,
                                const gchar *service_type,
                                const gchar *option);
```

Adds a request for service of `service_type` to the disco:Query being built.

Parameters

discovery	a LassoDiscovery	
service_type	requested service type	
option	option to the requested service	

Returns

0 if successful

lasso_discovery_process_request_msg ()

```
lasso_error_t
lasso_discovery_process_request_msg (LassoDiscovery *discovery,
                                     const gchar *message,
                                     const gchar *security_mech_id);
```

Process a received SOAP message for the discovery service.

Parameters

discovery	a LassoDiscovery object	
message	a serialized SOAP message	
security_mech_id	the security mech id to use for validating authorizations.	*[allow-none]*

Returns

0 if successfull, an error code otherwise.

lasso_discovery_build_response_msg ()

```
lasso_error_t
lasso_discovery_build_response_msg (LassoDiscovery *discovery);
```

Execute needed action for the received request, and produce a response message.

If any critical error occur, it tries to produce a SOAP fault.

Parameters

discovery	a LassoDiscovery object

Returns

0 on success; or an error code otherwise.

lasso_discovery_process_modify_response_msg ()

```
lasso_error_t
lasso_discovery_process_modify_response_msg
                               (LassoDiscovery *discovery,
                                const gchar *message);
```

Processes a disco:ModifyResponse SOAP message.

Parameters

discovery	a LassoDiscovery	
message	the disco:ModifyResponse SOAP message	

Returns

0 on success; or a negative value otherwise.

lasso_discovery_process_query_response_msg ()

```
lasso_error_t
lasso_discovery_process_query_response_msg
                              (LassoDiscovery *discovery,
                               const gchar *message);
```

Processes a disco:QueryResponse message. Extract credentials from the response and put them in the session, for later use by a request from a LassoWsfProfile.

Parameters

discovery	a LassoDiscovery	
message	the disco:QueryResponse message	

Returns

0 on success; or a negative value otherwise.

lasso_discovery_get_service ()

```
LassoWsfProfile~*
lasso_discovery_get_service (LassoDiscovery *discovery,
                             const char *service_type);
```

After a disco:query message, creates a LassoDataService instance for the requested `service_type` with the first resource offering found or the first resource offering matching the service type.

Parameters

discovery	a LassoDiscovery	
service_type	the requested service type	

Returns

a newly created LassoDataService object; or NULL if an error occured.

[transfer full][allow-none]

lasso_discovery_get_services ()

```
GList~*
lasso_discovery_get_services (LassoDiscovery *discovery);
```

After a disco:query message, creates a GList object of LassoDataService.

Parameters

discovery	a LassoDiscovery

Returns

a newly created GList object of LassoDataService; or NULL if an error occured.

[transfer full][element-type LassoNode]

LassoWsfProfileConstructor ()

```
LassoWsfProfile~*
(*LassoWsfProfileConstructor) (LassoServer *server,
                               LassoDiscoResourceOffering *offering);
```

lasso_discovery_register_constructor_for_service_type ()

```
void
lasso_discovery_register_constructor_for_service_type
                               (gchar const *service_type,
                                LassoWsfProfileConstructor constructor);
```

This function permits to subclass of LassoWsfProfile to register a constructor for the service type they supports.

Parameters

service_type	the URI of the service type	
constructor	a constructor function for the profile handling this service type	

lasso_discovery_unregister_constructor_for_service_type ()

```
void
lasso_discovery_unregister_constructor_for_service_type
                               (gchar const *service_type,
                                LassoWsfProfileConstructor constructor);
```

This function permits to subclass of LassoWsfProfile to unregister a constructor for the service type they previously registered using lasso_discovery_register_constructor_for_service_type().

Parameters

service_type	the URI of the service type	
constructor	a constructor function for the profile handling this service type	

Types and Values

struct LassoDiscovery

```
struct LassoDiscovery {
  LassoWsfProfile parent;

  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;
};
```

8.2 LassoDataService

LassoDataService — ID-WSF Data Service profile

Functions

LassoDataService *	lasso_data_service_new ()
LassoDataService *	lasso_data_service_new_full ()
lasso_error_t	lasso_data_service_init_query ()
lasso_error_t	lasso_data_service_add_query_item ()
lasso_error_t	lasso_data_service_build_modify_response_msg ()
lasso_error_t	lasso_data_service_build_response_msg ()
lasso_error_t	lasso_data_service_process_query_response_msg ()
lasso_error_t	lasso_data_service_get_answer ()
lasso_error_t	lasso_data_service_get_answers ()
lasso_error_t	lasso_data_service_get_answers_by_select ()
lasso_error_t	lasso_data_service_get_answers_by_item_id ()
lasso_error_t	lasso_data_service_init_modify ()
lasso_error_t	lasso_data_service_add_modification ()
lasso_error_t	lasso_data_service_process_modify_response_msg ()
lasso_error_t	lasso_data_service_process_request_msg ()
lasso_error_t	lasso_data_service_validate_request ()
lasso_error_t	lasso_data_service_build_query_response_msg ()
lasso_error_t	lasso_data_service_get_query_item ()
void	lasso_data_service_set_resource_data ()
xmlNode *	lasso_data_service_get_resource_data ()

Types and Values

struct	LassoDataService

Description

DataService allows Attribute Consumers (WSC) to request an Attribute Provider (WSP) to get or modify data about users with their consent.

Following up on LassoDiscovery first example, it created a *service* object, this is a LassoDataService instance. This example continues from that step and retrieves the name of the principal:

```
char *soap_answer;            // SOAP answer from data service
xmlNode *principal_name;      // libxml2 xmlNode with the principal name
```

```
service = lasso_discovery_get_service(discovery);
lasso_data_service_init_query(service, "/pp:PP/pp:InformalName", NULL);
lasso_data_service_build_request_msg(service);

// service must perform SOAP call to LASSO_WSF_PROFILE(service)->msg_url
// the SOAP message is LASSO_WSF_PROFILE(service)->msg_body.  The answer
// is stored in char* soap_answer;

lasso_data_service_process_query_response_msg(service, soap_answer);
principal_name = lasso_data_service_get_answer(service, "/pp:PP/pp:InformalName");

// app should probably then use xmlNodeGetContent libxml2 function to get
// access to node content.
```

Functions

lasso_data_service_new ()

```
LassoDataService~*
lasso_data_service_new (LassoServer *server);
```

Creates a new LassoDataService.

Parameters

server	the LassoServer

Returns

a newly created LassoDataService object; or NULL if an error occured.

lasso_data_service_new_full ()

```
LassoDataService~*
lasso_data_service_new_full (LassoServer *server,
                             LassoDiscoResourceOffering *offering);
```

Creates a new LassoDataService.

Parameters

server	the LassoServer
offering	the LassoDiscoResourceOffering

Returns

a newly created LassoDataService object; or NULL if an error occured.

lasso_data_service_init_query ()

```
lasso_error_t
lasso_data_service_init_query (LassoDataService *service,
                               const char *select,
                               const char *item_id,
                               const char *security_mech_id);
```

Initializes a new dst:Query request, asking for element `select` (with optional itemID set to `item_id`). `item_id` may be NULL only if the query won't contain other query items. You must follow this constraint, it will not be checked.

If both `select` and `item_id` are NULL, only a skeleton request is created and calls to lasso_data_service_add_query_item() will need to be done.

Parameters

service	a LassoDataService	
select	resource selection string (typically a XPath query)	
item_id	query item identifier (optional).	*[allow-none]*
security_mech_id	a security mechanism id.	*[allow-none]*

Returns

0 on success; or a negative value otherwise.

lasso_data_service_add_query_item ()

```
lasso_error_t
lasso_data_service_add_query_item (LassoDataService *service,
                                   const char *select,
                                   const char *item_id);
```

Adds a dst:QueryItem to the current dst:Query request. If there are already query item in the request and `itemId` is NULL, the call will fail.

Parameters

service	a LassoDataService	
select	resource selection string (typically a XPath query)	
item_id	query item identifier	

Returns

0 if sucessfull, an error code otherwise.

lasso_data_service_build_modify_response_msg ()

```
lasso_error_t
lasso_data_service_build_modify_response_msg
                              (LassoDataService *service);
```

lasso_data_service_build_response_msg ()

```
lasso_error_t
lasso_data_service_build_response_msg (LassoDataService *service);
```

Builds a dst:QueryResponse message.

Parameters

service	a LassoDataService	

Returns

0 on success; or a negative value otherwise.

lasso_data_service_process_query_response_msg ()

```
lasso_error_t
lasso_data_service_process_query_response_msg
                                        (LassoDataService *service,
                                         const char *message);
```

Processes a dst:Query message. Rebuilds a request object from the message and extracts ResourcedID.

Parameters

service	a LassoDataService	
message	the dst query response message	

Returns

0 on success; or a negative value otherwise.

lasso_data_service_get_answer ()

```
lasso_error_t
lasso_data_service_get_answer (LassoDataService *service,
                               xmlNode **output);
```

Get the first xmlNode of the first Data element of the QueryResponse message.

Parameters

service	a LassoDataService object.	
output	an xmlNode** pointer where to put the xmlNode* of the result.	*[out]*

Returns

0 if sucessful, an error code otherwise.

lasso_data_service_get_answers ()

```
lasso_error_t
lasso_data_service_get_answers (LassoDataService *service,
                                GList **output);
```

Get all the xmlNode content of the first Data element of the QueryResponse message.

Parameters

service	a LassoDataService object.	
output	an xmlNode** pointer where to put the xmlNode* of the result.	*[transfer full][allow-none][element-type xmlNode]*

Returns

0 if sucessful, an error code otherwise.

lasso_data_service_get_answers_by_select ()

```
lasso_error_t
lasso_data_service_get_answers_by_select
                               (LassoDataService *service,
                                const char *select,
                                GList **output);
```

Returns the answers for the specified `select` request.

Parameters

service	a LassoDataService	
select	resource selection string (typically a XPath query)	
output	a GList** to store a GList* containing the result, it must be freed.	*[allow-none][element-type xmlNode]*

Returns

0 if successful, an error code otheriwse

lasso_data_service_get_answers_by_item_id ()

```
lasso_error_t
lasso_data_service_get_answers_by_item_id
                               (LassoDataService *service,
                                const char *item_id,
                                GList **output);
```

Returns the answers for the specified `itemID` request.

Parameters

service	a LassoDataService	
item_id	query item identifier	
output	a GList** to store a GList* containing the result, it must be freed.	*[allow-none][element-type xmlNode]*

Returns

0 if successful, an error code otherwise

lasso_data_service_init_modify ()

```
lasso_error_t
lasso_data_service_init_modify (LassoDataService *service,
                                const char *security_mech_id);
```

Initialize a Data Service Template Modify request using a command to select some data, and an XML fragment to replace the selected data.

Parameters

service	a LassoDataService object	
security_mech_id	a security mechanism id.	*[allow-none]*

Returns

0 if successful, an error code otherwise.

lasso_data_service_add_modification ()

```
lasso_error_t
lasso_data_service_add_modification (LassoDataService *service,
                                     const gchar *select,
                                     xmlNode *xmlData,
                                     gboolean overrideAllowed,
                                     time_t *notChangedSince,
                                     LassoDstModification **output);
```

Add a new modification to the current modify request. If overrideAllowed is FALSE, xmlData must absolutely be present. Refer to the ID-WSF DST 1.0 specification for the semantic of the created message.

Parameters

service	a LassoDataService object	
select	a selector string	
xmlData	optional NewData content.	*[allow-none]*
overrideAllowed	whether to permit delete or replace of existings.	*[allow-none][default FALSE]*

notChangedSince	if not NULL, give the time (as a local time_t value) of the last known modification to the datas, it is used to permit secure concurrent accesses.	*[allow-none]*
output	a LassoDstModification** pointer where to put the LassoDstModification of the result.	*[out]*

Returns

0 if successful and the new modification object in *output, an error code otherwise.

lasso_data_service_process_modify_response_msg ()

```
lasso_error_t
lasso_data_service_process_modify_response_msg
                               (LassoDataService *service,
                                const gchar *soap_msg);
```

Process a modify response message.

Parameters

service	a LassoDataService	
soap_msg	the SOAP message	

Returns

0 on success; or a negative value otherwise.

lasso_data_service_process_request_msg ()

```
lasso_error_t
lasso_data_service_process_request_msg
                               (LassoDataService *service,
                                const char *message,
                                const char *security_mech_id);
```

Parameters

service	a LassoDataService object	
message	a C string containing the SOAP request	
security_mech_id	a C string describing the required security mechanism or NULL.	*[allow-none]*

Returns

0 if successfull, an error code otherwise.

lasso_data_service_validate_request ()

```
lasso_error_t
lasso_data_service_validate_request (LassoDataService *service);
```

lasso_data_service_build_query_response_msg ()

```
lasso_error_t
lasso_data_service_build_query_response_msg
                               (LassoDataService *service);
```

lasso_data_service_get_query_item ()

```
lasso_error_t
lasso_data_service_get_query_item (LassoDataService *service,
                                   const char *select,
                                   const char *item_id,
                                   LassoDstQueryItem **output);
```

Look up the first query item in the current request matching the given criteria, `select` or `item_id` . At least one of the criteria must be present for the call to succeed.

Parameters

service	a LassoDataService	
select	the select string of the query item to found	
item_id	the item id of the query item to found	
output	a LassoDstQueryItem handle to store the result object, its reference count is not incremented.	*[transfer none]*

Returns

0 if successful, an error code otherwise.

lasso_data_service_set_resource_data ()

```
void
lasso_data_service_set_resource_data (LassoDataService *service,
                                      const xmlNode *resource_data);
```

Set the resource data content.

Parameters

service	a LassoDataService object	
resource_data	an xmlnode representing the resource data of the service.	*[allow-none]*

lasso_data_service_get_resource_data ()

```
xmlNode~*
lasso_data_service_get_resource_data (LassoDataService *service);
```

Return the XML resrouce data in this data service.

Parameters

service	a LassoDataService object	

Returns

a newly allocated xmlNode or NULL.

[transfer full][allow-none]

Types and Values

struct LassoDataService

```
struct LassoDataService {
  LassoWsfProfile parent;
};
```

8.3 LassoPersonalProfileService

LassoPersonalProfileService — a subclass of LassoDataService to access Personal Profile datas

Stability Level

Unstable, unless otherwise indicated

Functions

LassoPersonalProfileService *	lasso_personal_profile_service_new ()
LassoPersonalProfileService *	lasso_personal_profile_service_new_full ()
gchar *	lasso_personal_profile_service_get_email ()

Types and Values

struct	LassoPersonalProfileService

Description

Functions

lasso_personal_profile_service_new ()

```
LassoPersonalProfileService~*
lasso_personal_profile_service_new (LassoServer *server);
```

lasso_personal_profile_service_new_full ()

```
LassoPersonalProfileService~*
lasso_personal_profile_service_new_full
                               (LassoServer *server,
                                LassoDiscoResourceOffering *offering);
```

lasso_personal_profile_service_get_email ()

```
gchar~*
lasso_personal_profile_service_get_email
                               (LassoPersonalProfileService *service);
```

Types and Values

struct LassoPersonalProfileService

```
struct LassoPersonalProfileService {
  LassoDataService parent;
};
```

8.4 LassoAuthentication

LassoAuthentication —

Functions

LassoAuthentication *	lasso_authentication_new ()
void	lasso_authentication_destroy ()
lasso_error_t	lasso_authentication_client_start ()
lasso_error_t	lasso_authentication_client_step ()
char *	lasso_authentication_get_mechanism_list ()
lasso_error_t	lasso_authentication_init_request ()
lasso_error_t	lasso_authentication_process_request_msg ()
lasso_error_t	lasso_authentication_process_response_msg ()
lasso_error_t	lasso_authentication_server_start ()
lasso_error_t	lasso_authentication_server_step ()

Types and Values

typedef	LassoUserAccount
enum	LassoSaslMechanisms
struct	LassoAuthentication

Description

Functions

lasso_authentication_new ()

```
LassoAuthentication~*
lasso_authentication_new (LassoServer *server);
```

lasso_authentication_destroy ()

```
void
lasso_authentication_destroy (LassoAuthentication *authentication);
```

lasso_authentication_client_start ()

```
lasso_error_t
lasso_authentication_client_start (LassoAuthentication *authentication);
```

lasso_authentication_client_step ()

```
lasso_error_t
lasso_authentication_client_step (LassoAuthentication *authentication);
```

lasso_authentication_get_mechanism_list ()

```
char~*
lasso_authentication_get_mechanism_list
                               (LassoAuthentication *authentication);
```

lasso_authentication_init_request ()

```
lasso_error_t
lasso_authentication_init_request (LassoAuthentication *authentication,
                                   LassoDiscoDescription *description,
                                   const gchar *mechanisms,
                                   LassoUserAccount *account);
```

lasso_authentication_process_request_msg ()

```
lasso_error_t
lasso_authentication_process_request_msg
                               (LassoAuthentication *authentication,
                                const gchar *soap_msg);
```

lasso_authentication_process_response_msg ()

```
lasso_error_t
lasso_authentication_process_response_msg
                               (LassoAuthentication *authentication,
                                const gchar *soap_msg);
```

lasso_authentication_server_start ()

```
lasso_error_t
lasso_authentication_server_start (LassoAuthentication *authentication);
```

lasso_authentication_server_step ()

```
lasso_error_t
lasso_authentication_server_step (LassoAuthentication *authentication);
```

Types and Values

LassoUserAccount

```
typedef struct LassoUserAccount LassoUserAccount;
```

enum LassoSaslMechanisms

Members

LASSO_SASL_MECH_ANONYMOUS		
LASSO_SASL_MECH_PLAIN		
LASSO_SASL_MECH_CRAM_MD5		

struct LassoAuthentication

```
struct LassoAuthentication {
  LassoWsfProfile parent;

  /* The SASL context kept for the life of the connection */
  sasl_conn_t *connection;
  sasl_interact_t **client_interact;
};
```

8.5 LassoWsfProfile

LassoWsfProfile — Base class for ID-WSF 1.0 services

Stability Level

Unstable, unless otherwise indicated

Functions

LassoIdentity *	lasso_wsf_profile_get_identity ()
LassoSession *	lasso_wsf_profile_get_session ()
gboolean	lasso_wsf_profile_is_identity_dirty ()
gboolean	lasso_wsf_profile_is_session_dirty ()
lasso_error_t	lasso_wsf_profile_set_identity_from_dump ()
lasso_error_t	lasso_wsf_profile_set_session_from_dump ()
lasso_error_t	lasso_wsf_profile_build_soap_request_msg ()
lasso_error_t	lasso_wsf_profile_build_soap_response_msg ()
lasso_error_t	lasso_wsf_profile_init_soap_request ()
lasso_error_t	lasso_wsf_profile_init_soap_response ()
lasso_error_t	lasso_wsf_profile_process_soap_request_msg ()
lasso_error_t	lasso_wsf_profile_process_soap_response_msg ()
LassoWsfProfile *	lasso_wsf_profile_new ()
LassoWsfProfile *	lasso_wsf_profile_new_full ()
lasso_error_t	lasso_wsf_profile_set_description_from_offering ()
void	lasso_wsf_profile_set_description ()
LassoDiscoDescription *	lasso_wsf_profile_get_description ()
LassoDiscoResourceOffering *	lasso_wsf_profile_get_resource_offering ()
lasso_error_t	lasso_wsf_profile_set_security_mech_id ()
const char *	lasso_wsf_profile_get_security_mech_id ()
lasso_error_t	lasso_wsf_profile_init ()
lasso_error_t	lasso_wsf_profile_get_remote_provider ()
const char *	lasso_wsf_profile_get_remote_provider_id ()
LassoSoapFault *	lasso_wsf_profile_get_soap_fault ()
lasso_error_t	lasso_wsf_profile_set_soap_fault ()
lasso_error_t	lasso_wsf_profile_set_status_code ()
const char *	lasso_wsf_profile_get_status_code ()
lasso_error_t	lasso_wsf_profile_set_msg_url_from_description ()
void	lasso_wsf_profile_set_resource_offering ()
lasso_error_t	lasso_wsf_profile_init_interaction_service_redirect ()

Types and Values

struct	LassoWsfProfile

Description

Use this class to base your ID-WSF 1.0 services.

Functions

lasso_wsf_profile_get_identity ()

```
LassoIdentity~*
lasso_wsf_profile_get_identity (const LassoWsfProfile *profile);
```

Warning

`lasso_wsf_profile_get_identity` is deprecated and should not be used in newly-written code.

Gets the identity bound to *profile* .

Parameters

profile	a LassoWsfProfile

Returns

the identity or NULL if it none was found. The LassoIdentity object is internally allocated and must not be freed by the caller. *[transfer none][allow-none]*

lasso_wsf_profile_get_session ()

```
LassoSession~*
lasso_wsf_profile_get_session (const LassoWsfProfile *profile);
```

Gets the session bound to *profile* .

Parameters

profile	a LassoWsfProfile

Returns

the session or NULL if it none was found. The LassoSession object is internally allocated and must not be freed by the caller. *[transfer none][allow-none]*

lasso_wsf_profile_is_identity_dirty ()

```
gboolean
lasso_wsf_profile_is_identity_dirty (const LassoWsfProfile *profile);
```

Checks whether identity has been modified (and should therefore be saved).

Parameters

profile	a LassoWsfProfile

Returns

TRUE if identity has changed

lasso_wsf_profile_is_session_dirty ()

```
gboolean
lasso_wsf_profile_is_session_dirty (const LassoWsfProfile *profile);
```

Checks whether session has been modified (and should therefore be saved).

Parameters

profile	a LassoWsfProfile

Returns

TRUE if session has changed

lasso_wsf_profile_set_identity_from_dump ()

```
lasso_error_t
lasso_wsf_profile_set_identity_from_dump
                               (LassoWsfProfile *profile,
                                const gchar *dump);
```

Builds a new LassoIdentity object from XML dump and binds it to *profile* .

Parameters

profile	a LassoWsfProfile
dump	XML identity dump

Returns

0 on success; or a negative value otherwise.

lasso_wsf_profile_set_session_from_dump ()

```
lasso_error_t
lasso_wsf_profile_set_session_from_dump
                               (LassoWsfProfile *profile,
                                const gchar *dump);
```

Builds a new LassoSession object from XML dump and binds it to *profile* .

Parameters

profile	a LassoWsfProfile
dump	XML session dump

Returns

0 on success; or a negative value otherwise.

lasso_wsf_profile_build_soap_request_msg ()

```
lasso_error_t
lasso_wsf_profile_build_soap_request_msg
                               (LassoWsfProfile *profile);
```

Warning
`lasso_wsf_profile_build_soap_request_msg` is deprecated and should not be used in newly-written code.

Create the char* string containing XML document for the SOAP ID-WSF request and eventually sign with the local public depending on the security mechanism requested.

Parameters

profile	the LassoWsfProfile object

Returns

0 if construction is successfull.

lasso_wsf_profile_build_soap_response_msg ()

```
lasso_error_t
lasso_wsf_profile_build_soap_response_msg
                               (LassoWsfProfile *profile);
```

Create the char* string containing XML document for the SOAP ID-WSF response.

Parameters

wsf_profile	a LassoWsfProfile object

Returns

0 if construction is successfull.

lasso_wsf_profile_init_soap_request ()

```
lasso_error_t
lasso_wsf_profile_init_soap_request (LassoWsfProfile *profile,
                                     LassoNode *request);
```

Build the SOAP envelope for a request to and ID-WSF 1.0 web service and set the body of the request to request. The reference to request is not stolen i.e the ref count of request is increased by one after this call.

Parameters

profile	a LassoWsfProfile to initialize for a SOAP request
request	a LassoNode object containing the body for the SOAP request, can be NULL.

Returns

0 if initialization was successfull.

lasso_wsf_profile_init_soap_response ()

```
lasso_error_t
lasso_wsf_profile_init_soap_response (LassoWsfProfile *profile,
                                      LassoNode *response);
```

Build a new SOAP envelope containing response to current SOAP request

Parameters

profile	a LassoWsfProfile object	
response	a LassoNode object	

lasso_wsf_profile_process_soap_request_msg ()

```
lasso_error_t
lasso_wsf_profile_process_soap_request_msg
                               (LassoWsfProfile *profile,
                                const gchar *message,
                                const gchar *security_mech_id);
```

Process an ID-WSF SOAP request, extract headers information, and check compliance with the security mechanism.

Parameters

profile	a LassoWsfProfile object	
message	a SOAP request message string	
security_mech_id	the security mechanism to apply	

Returns

0 if successful, an error code otherwise.

lasso_wsf_profile_process_soap_response_msg ()

```
lasso_error_t
lasso_wsf_profile_process_soap_response_msg
                               (LassoWsfProfile *profile,
                                const gchar *message);
```

Parse a SOAP response from an ID-WSF 1.0 service, eventually signal a SOAP fault.

Parameters

profile	a LassoWsfProfile object	
message	the textual representaition of a SOAP message	

Returns

0 if the processing of this message was successful.

lasso_wsf_profile_new ()

```
LassoWsfProfile~*
lasso_wsf_profile_new (LassoServer *server);
```

Warning
`lasso_wsf_profile_new` is deprecated and should not be used in newly-written code.

Create a new WsfProfile with the given LassoServer object.

Return: a new LassoWsfProfile if creation and initialization were successfull, NULL otherwise.

Parameters

server	a LassoServer object to lookup remote provider informations

lasso_wsf_profile_new_full ()

```
LassoWsfProfile~*
lasso_wsf_profile_new_full (LassoServer *server,
                            LassoDiscoResourceOffering *offering);
```

Create a new WsfProfile with the given LassoServer object and the given LassoDiscoResourceOffering.

Return: a new LassoWsfProfile if creation and initialization were successfull, NULL otherwise.

Parameters

server	a LassoServer object to lookup remote provider informations.
offering	a LassoDiscoResourceOffering for the requested service.

lasso_wsf_profile_set_description_from_offering ()

```
lasso_error_t
lasso_wsf_profile_set_description_from_offering
                               (LassoWsfProfile *profile,
                                const LassoDiscoResourceOffering *offering,
                                const char *security_mech_id);
```

Warning
`lasso_wsf_profile_set_description_from_offering` is deprecated and should not be used in newly-written code.

Setup the LassoWsfProfile for a given security mechanism.

Parameters

profile	a LassoWsfProfile	
offering	a LassoDiscoResourceOffering containing descriptions	
security_mech_id	an URL representing the wished security mechanism, if NULL take the first descriptions	

Returns

0 if a corresponding description was found, LASSO_PROFILE_ERROR_MISSING_SERVICE_DESCRIPTION if no description with the given security mechanism was found.

lasso_wsf_profile_set_description ()

```
void
lasso_wsf_profile_set_description (LassoWsfProfile *profile,
                                   LassoDiscoDescription *description);
```

Set the currently registered LassoDiscoDescription, that permits to locate the endpoint and the security mechanism to use for the next ID-WSF request.

Parameters

profile	the LassoWsfProfile	
description	a LassoDiscoDescription	

lasso_wsf_profile_get_description ()

```
LassoDiscoDescription~*
lasso_wsf_profile_get_description (const LassoWsfProfile *profile);
```

Returns the currently registered LassoDiscoDescription, that permits to locate the endpoint and the security mechanism to use for the next ID-WSF request.

Parameters

profile	a LassoWsfProfile	

Returns

a LassoDiscoDescriptio or NULL if none is present.

[transfer full]

lasso_wsf_profile_get_resource_offering ()

```
LassoDiscoResourceOffering~*
lasso_wsf_profile_get_resource_offering
                                (LassoWsfProfile *profile);
```

Returns the ResourceOffering setupt with this profile object.

Parameters

profile	the LassoWsfProfile object

Returns

a LassoDiscoResourceOffering if one was setup during construction, NULL otherwise.

[transfer full]

lasso_wsf_profile_set_security_mech_id ()

```
lasso_error_t
lasso_wsf_profile_set_security_mech_id
                                (LassoWsfProfile *profile,
                                 const char *security_mech_id);
```

Set the security mechanism to use. Currently only SAML and NULL mechanism are supported for authentication. Transport is not handled by lasso so all are supported.

List of supported mechanism ids: LASSO_SECURITY_MECH_NULL or "urn:liberty:security:2003-08:null:null" LASSO_SECURITY
or "urn:liberty:security:2003-08:null:SAML" LASSO_SECURITY_MECH_TLS or "urn:liberty:security:2003-08:TLS:null" LASSO_S
or "urn:liberty:security:2003-08:TLS:SAML" LASSO_SECURITY_MECH_CLIENT_TLS or "urn:liberty:security:2003-08:ClientTLS
LASSO_SECURITY_MECH_CLIENT_TLS_SAML or "urn:liberty:security:2003-08:ClientTLS:SAML"

Parameters

profile	the LassoWsfProfile object
security_mech_id	a char* string representing the chosen security mech id.

Returns

0 if the security mechanism is supported by this LassoWsfProfile object, an error code otherwise.

lasso_wsf_profile_get_security_mech_id ()

```
const char~*
lasso_wsf_profile_get_security_mech_id
                                (LassoWsfProfile *profile);
```

Parameters

profile	the LassoWsfProfile object	

Returns

the current security mechanism id for this object.

[transfer none]

lasso_wsf_profile_init ()

```
lasso_error_t
lasso_wsf_profile_init (LassoWsfProfile *profile,
                        LassoServer *server,
                        LassoDiscoResourceOffering *offering);
```

Initialize a LassoWsfProfile in order to handle or send request to, an ID-WSF web service.

Return: 0 if initialization was successfull.

Parameters

profile	the LassoWsfProfile to initialize	
server	a LassoServer object to resolve provider IDs.	
offering	a LassoDiscoResourceOffering for the targetted web service.	

lasso_wsf_profile_get_remote_provider ()

```
lasso_error_t
lasso_wsf_profile_get_remote_provider (LassoWsfProfile *wsf_profile,
                                       LassoProvider **provider);
```

Return the remote provider parsed in the last processing of a SOAP request or a SOAP response.

Parameters

wsf_profile	a LassoWsfProfile object	
provider	an output pointer to LassoProvider object variable.	*[transfer full]*

Returns

0 if successfull, LASSO_PROFILE_ERROR_MISSING_REMOTE_PROVIDERID if no provider id is present in the SOAP headers, or LASSO_SERVER_ERROR_PROVIDER_NOT_FOUND if the provider is unknown to us.

lasso_wsf_profile_get_remote_provider_id ()

```
const char~*
lasso_wsf_profile_get_remote_provider_id
                               (LassoWsfProfile *wsf_profile);
```

Return the remote provider id parsed in the last processing of a SOAP request or a SOAP response.

Parameters

wsf_profile	a LassoWsfProfile object

Returns

the provider id string or NULL.

[transfer none][allow-none]

lasso_wsf_profile_get_soap_fault ()

```
LassoSoapFault~*
lasso_wsf_profile_get_soap_fault (LassoWsfProfile *wsf_profile);
```

lasso_wsf_profile_set_soap_fault ()

```
lasso_error_t
lasso_wsf_profile_set_soap_fault (LassoWsfProfile *wsf_profile,
                                  LassoSoapFault *soap_fault);
```

Set a SOAP fault to be returned in next SOAP response message. The SOAP fault is removed by lasso_wsf_profile_init_soap_request.

Parameters

wsf_profile	a LassoWsfProfile object
soap_fault	a LassoSoapFault object

Returns

0 if successful, an error code otherwise.

lasso_wsf_profile_set_status_code ()

```
lasso_error_t
lasso_wsf_profile_set_status_code (LassoWsfProfile *wsf_profile,
                                   const char *code);
```

Set the status code to set in the next built response.

Parameters

wsf_profile	a LassoWsfProfile
status_code	a string representing the status code

Returns

0 if successful, an error code otherwise.

lasso_wsf_profile_get_status_code ()

```
const char~*
lasso_wsf_profile_get_status_code (LassoWsfProfile *wsf_profile);
```

Return the actual status code for this protocol object.

Parameters

wsf_profile	a LassoWsfProfile object

Returns

a string owned by the profile object or NULL.

[transfer none][allow-none]

lasso_wsf_profile_set_msg_url_from_description ()

```
lasso_error_t
lasso_wsf_profile_set_msg_url_from_description
                               (LassoWsfProfile *wsf_profile);
```

lasso_wsf_profile_set_resource_offering ()

```
void
lasso_wsf_profile_set_resource_offering
                               (LassoWsfProfile *profile,
                                LassoDiscoResourceOffering *offering);
```

Set the Resssource Offering to setup this ID-WSF profile.

Parameters

profile	a LassoWsfProfile
offering	a LassoDiscoResourceOffering

lasso_wsf_profile_init_interaction_service_redirect ()

```
lasso_error_t
lasso_wsf_profile_init_interaction_service_redirect
                               (LassoWsfProfile *profile,
                                char *redirect_url);
```

Types and Values

struct LassoWsfProfile

```
struct LassoWsfProfile {
  LassoNode parent;

  LassoServer *server;

  LassoNode *request;
  LassoNode *response;

  LassoSoapEnvelope *soap_envelope_request;
  LassoSoapEnvelope *soap_envelope_response;

  gchar *msg_url;
  gchar *msg_body;
};
```

8.6 LassoInteractionProfileService

LassoInteractionProfileService — A service to request user interaction from a principal

Stability Level

Unstable, unless otherwise indicated

Functions

LassoInteractionProfileService *	lasso_interaction_profile_service_new ()
lasso_error_t	lasso_interaction_profile_service_init_request ()
lasso_error_t	lasso_interaction_profile_service_process_request_msg ()
lasso_error_t	lasso_interaction_profile_service_process_response_msg ()

Types and Values

struct	LassoInteractionProfileService

Description

Functions

lasso_interaction_profile_service_new ()

```
LassoInteractionProfileService~*
lasso_interaction_profile_service_new (LassoServer *server);
```

lasso_interaction_profile_service_init_request ()

```
lasso_error_t
lasso_interaction_profile_service_init_request
                               (LassoInteractionProfileService *service);
```

lasso_interaction_profile_service_process_request_msg ()

```
lasso_error_t
lasso_interaction_profile_service_process_request_msg
                               (LassoInteractionProfileService *service,
                                const gchar *request_msg);
```

lasso_interaction_profile_service_process_response_msg ()

```
lasso_error_t
lasso_interaction_profile_service_process_response_msg
                               (LassoInteractionProfileService *service,
                                const gchar *response_msg);
```

Types and Values

struct LassoInteractionProfileService

```
struct LassoInteractionProfileService {
  LassoWsfProfile parent;
};
```

8.7 Utility functions for ID-WSF 1.0

Utility functions for ID-WSF 1.0 — Misc functions used in the implementation of ID-WSF 1.0

Stability Level

Private, unless otherwise indicated

Functions

gboolean	lasso_security_mech_id_is_null_authentication ()
gboolean	lasso_security_mech_id_is_x509_authentication ()
gboolean	lasso_security_mech_id_is_saml_authentication ()
gboolean	lasso_security_mech_id_is_bearer_authentication ()
lasso_error_t	lasso_login_set_encryptedResourceId ()
lasso_error_t	lasso_login_set_resourceId ()
LassoDiscoServiceInstance *	lasso_server_get_service ()
lasso_error_t	lasso_server_add_service ()
lasso_error_t	lasso_server_add_service_from_dump ()
lasso_error_t	lasso_identity_add_resource_offering ()
gboolean	lasso_identity_remove_resource_offering ()
GList *	lasso_identity_get_offerings ()

LassoDiscoResourceOffering *	lasso_identity_get_resource_offering ()
LassoSoapBindingProvider *	lasso_soap_binding_get_provider ()
LassoSoapBindingCorrelation *	lasso_soap_binding_get_correlation ()
void	lasso_register_dst_service ()
char *	lasso_get_prefix_for_dst_service_href ()

Description

Functions

lasso_security_mech_id_is_null_authentication ()

```
gboolean
lasso_security_mech_id_is_null_authentication
                                (const gchar *security_mech_id);
```

Parameters

security_mech_id	the URI of an authentication mechanism

Returns

TRUE if *security_mech_id* is null or one of urn:liberty:security:2003-08:null:null, urn:liberty:security:2003-08:TLS:null, urn:liberty:security:2003-08:ClientTLS:null, FALSE otherwise.

lasso_security_mech_id_is_x509_authentication ()

```
gboolean
lasso_security_mech_id_is_x509_authentication
                                (const gchar *security_mech_id);
```

Parameters

security_mech_id	the URI of an authentication mechanism

Returns

TRUE if *security_mech_id* is one of urn:liberty:security:2003-08:null:X509, urn:liberty:security:2003-08:TLS:X509, urn:liberty:sec
08:ClientTLS:X509, FALSE otherwise.

lasso_security_mech_id_is_saml_authentication ()

```
gboolean
lasso_security_mech_id_is_saml_authentication
                                (const gchar *security_mech_id);
```

Parameters

security_mech_id	the URI of an authentication mechanism

Returns

TRUE if *security_mech_id* is one of urn:liberty:security:2003-08:null:SAML, urn:liberty:security:2003-08:TLS:SAML, urn:liberty:s
08:ClientTLS:SAML, urn:liberty:security:2005-02:null:SAML, urn:liberty:security:2005-02:TLS:SAML, FALSE otherwise.

lasso_security_mech_id_is_bearer_authentication ()

```
gboolean
lasso_security_mech_id_is_bearer_authentication
                               (const gchar *security_mech_id);
```

Parameters

security_mech_id	the URI of an authentication mechanism

Returns

TRUE if *security_mech_id* is one of urn:liberty:security:2003-08:null:Bearer, urn:liberty:security:2003-08:TLS:Bearer, urn:liberty:s
08:ClientTLS:Bearer, urn:liberty:security:2005-02:null:Bearer, urn:liberty:security:2005-02:TLS:Bearer, FALSE otherwise.

lasso_login_set_encryptedResourceId ()

```
lasso_error_t
lasso_login_set_encryptedResourceId (LassoLogin *login,
                                     LassoDiscoEncryptedResourceID *encryptedResourceId);
```

Set the LassoDiscoEncryptedResourceID to place the next produced assertions as an ID-WSF 1.0 bootstrap.

Parameters

login	a LassoLogin object
encryptedResourceId	the LassoDiscoEncryptedResourceID to setup in the login object

Returns

0 on success; or a negative value otherwise.

lasso_login_set_resourceId ()

```
lasso_error_t
lasso_login_set_resourceId (LassoLogin *login,
                            const char *content);
```

Set the resourceId to place in the next produced assertion for ID-WSF bootstrap.

Parameters

login	a LassoLogin	
content	a resourceID identifier	

Returns

0 on success; or a negative value otherwise.

lasso_server_get_service ()

```
LassoDiscoServiceInstance~*
lasso_server_get_service (LassoServer *server,
                          const gchar *serviceType);
```

Look up a disco service instance corresponding to this service type.

Parameters

server	a LassoServer	
serviceType	the service type	

Returns

the LassoDiscoServiceInstance, NULL if it was not found. The LassoDiscoServiceInstance is owned by Lasso and should not be freed.

[transfer none][allow-none]

lasso_server_add_service ()

```
lasso_error_t
lasso_server_add_service (LassoServer *server,
                          LassoNode *service);
```

Add a service to the registry of service of this LassoServer object.

Parameters

server	a LassoServer	
service	a LassoNode object implementing representing a service endpoint.	

Returns

0 on success; a negative value if an error occured.

lasso_server_add_service_from_dump ()

```
lasso_error_t
lasso_server_add_service_from_dump (LassoServer *server,
                                             const gchar *dump);
```

An utility function that parse a LassoNode dump an try to add it as a service using lasso_server_add_service.

Parameters

server	a LassoServer	
dump	the XML dump of a LassoNode representing a service endpoint.	

Returns

0 if succesfull, LASSO_PARAM_ERROR_BAD_TYPE_OF_NULL_OBJECT if said dump is not a LassoNode or is not of the righ type, LASSO_PARAM_ERROR_INVALID_VALUE if dump is NULL.

lasso_identity_add_resource_offering ()

```
lasso_error_t
lasso_identity_add_resource_offering (LassoIdentity *identity,
                                      LassoDiscoResourceOffering *offering);
```

Add a new offering to the identity object to be retrieved later by lasso_identity_get_offerings() or lasso_identity_get_resource_offering(). It also allocate an entryId identifier for the offering, look into offering->entryID to get it after this call.

Parameters

identity	a LassoIdentity object	
offering	a LassoDiscoResourceOffering object to add	

Returns

Always 0, there should not be any error (if memory is not exhausted).

lasso_identity_remove_resource_offering ()

```
gboolean
lasso_identity_remove_resource_offering
                               (LassoIdentity *identity,
                                const char *entryID);
```

Remove resource offering about identity with *entryID*

Parameters

identity	a LassoIdentity	
entryID	the resource offering entry ID	

Returns

TRUE on success; FALSE if the offering was not found.

lasso_identity_get_offerings ()

```
GList~*
lasso_identity_get_offerings (LassoIdentity *identity,
                              const char *service_type);
```

Returns a list of LassoDiscoResourceOffering associated to this service type.

Parameters

identity	a LassoIdentity
service_type	a char* string representing the type of service we are looking for

Returns

a newly allocated list of LassoDiscoResourceOffering.

[transfer full][element-type LassoDiscoResourceOffering]

lasso_identity_get_resource_offering ()

```
LassoDiscoResourceOffering~*
lasso_identity_get_resource_offering (LassoIdentity *identity,
                                      const char *entryID);
```

lasso_soap_binding_get_provider ()

```
LassoSoapBindingProvider~*
lasso_soap_binding_get_provider (LassoSoapEnvelope *envelope);
```

Look up the sb:Provider header in the SOAP message envelope.

Parameters

envelope	a LassoSoapEnvelope

Returns

NULL if no Provider element is present in the header of the SOAP envelope. If found it returns a reference you do not own.

lasso_soap_binding_get_correlation ()

```
LassoSoapBindingCorrelation~*
lasso_soap_binding_get_correlation (LassoSoapEnvelope *evelope);
```

Look up the sb:Correlation header in the SOAP message envelope.

Parameters

envelope	a LassoSoapEnvelope

Returns

NULL if no Correlation element is present in the header of the SOAP envelope. If found it returns a reference you do not own.

lasso_register_dst_service ()

```
void
lasso_register_dst_service (const char *prefix,
                            const char *href);
```

Registers prefix and href of a custom data service template service.

Parameters

prefix	prefix of DST service
href	href of DST service

lasso_get_prefix_for_dst_service_href ()

```
char~*
lasso_get_prefix_for_dst_service_href (const char *href);
```

Chapter 9

Objects from ID-WSF 1.0 schemas

9.1 ID-WSF 1.0 Strings

ID-WSF 1.0 Strings —

Types and Values

#define	LASSO_DISCO_HREF
#define	LASSO_DISCO_PREFIX
#define	LASSO_EP_HREF
#define	LASSO_EP_PREFIX
#define	LASSO_IS_HREF
#define	LASSO_IS_PREFIX
#define	LASSO_PP10_HREF
#define	LASSO_PP10_PREFIX
#define	LASSO_SA_HREF
#define	LASSO_SA_PREFIX
#define	LASSO_SEC_HREF
#define	LASSO_SEC_PREFIX
#define	LASSO_SOAP_BINDING_EXT_HREF
#define	LASSO_SOAP_BINDING_EXT_PREFIX
#define	LASSO_SOAP_BINDING_HREF
#define	LASSO_SOAP_BINDING_PREFIX
#define	LASSO_SECURITY_MECH_NULL
#define	LASSO_SECURITY_MECH_X509
#define	LASSO_SECURITY_MECH_SAML
#define	LASSO_SECURITY_MECH_BEARER
#define	LASSO_SECURITY_MECH_TLS
#define	LASSO_SECURITY_MECH_TLS_X509
#define	LASSO_SECURITY_MECH_TLS_SAML
#define	LASSO_SECURITY_MECH_TLS_BEARER
#define	LASSO_SECURITY_MECH_CLIENT_TLS
#define	LASSO_SECURITY_MECH_CLIENT_TLS_X509
#define	LASSO_SECURITY_MECH_CLIENT_TLS_SAML
#define	LASSO_SECURITY_MECH_CLIENT_TLS_BEARER
#define	LASSO_SECURITY11_MECH_X509
#define	LASSO_SECURITY11_MECH_SAML
#define	LASSO_SECURITY11_MECH_BEARER
#define	LASSO_SECURITY11_MECH_TLS_X509

#define	LASSO_SECURITY11_MECH_TLS_SAML
#define	LASSO_SECURITY11_MECH_TLS_BEARER
#define	LASSO_SA_SASL_SERVICE_NAME
#define	LASSO_IS_INTERACT_ATTR_INTERACT_IF_NEEDED
#define	LASSO_IS_INTERACT_ATTR_DO_NOT_INTERACT
#define	LASSO_IS_INTERACT_ATTR_DO_NOT_INTERACT_FOR_DATA
#define	LASSO_DISCO_STATUS_CODE_OK
#define	LASSO_DISCO_STATUS_CODE_DISCO_OK
#define	LASSO_DISCO_STATUS_CODE_FAILED
#define	LASSO_DISCO_STATUS_CODE_REMOVE_ENTRY
#define	LASSO_DISCO_STATUS_CODE_FORBIDDEN
#define	LASSO_DISCO_STATUS_CODE_NO_RESULTS
#define	LASSO_DISCO_STATUS_CODE_DIRECTIVES
#define	LASSO_DST_STATUS_CODE_OK
#define	LASSO_DST_STATUS_CODE_FAILED
#define	LASSO_DST_STATUS_CODE_PARTIAL
#define	LASSO_DST_STATUS_CODE_ACTION_NOT_AUTHORIZED
#define	LASSO_DST_STATUS_CODE_ACTION_NOT_SUPPORTED
#define	LASSO_DST_STATUS_CODE_ALL_RETURNED
#define	LASSO_DST_STATUS_CODE_CHANGE_HISTORY_NOT_SUPPORTED
#define	LASSO_DST_STATUS_CODE_CHANGED_SINCE_RETURNS_ALL
#define	LASSO_DST_STATUS_CODE_DATA_TOO_LONG
#define	LASSO_DST_STATUS_CODE_EXISTS_ALREADY
#define	LASSO_DST_STATUS_CODE_EXTENSION_NOT_SUPPORTED
#define	LASSO_DST_STATUS_CODE_INVALID_DATA
#define	LASSO_DST_STATUS_CODE_INVALID_RESOURCE_ID
#define	LASSO_DST_STATUS_CODE_INVALID_SELECT
#define	LASSO_DST_STATUS_CODE_MISSING_NEW_DATA_ELEMENT
#define	LASSO_DST_STATUS_CODE_MISSING_RESOURCE_ID_ELEMENT
#define	LASSO_DST_STATUS_CODE_MISSING_SELECT
#define	LASSO_DST_STATUS_CODE_MODIFIED_SINCE
#define	LASSO_DST_STATUS_CODE_NO_MORE_ELEMENTS
#define	LASSO_DST_STATUS_CODE_NO_MULTIPLE_ALLOWED
#define	LASSO_DST_STATUS_CODE_NO_MULTIPLE_RESOURCES
#define	LASSO_DST_STATUS_CODE_TIME_OUT
#define	LASSO_DST_STATUS_CODE_UNEXPECTED_ERROR
#define	LASSO_SA_STATUS_CODE_OK
#define	LASSO_SA_STATUS_CODE_CONTINUE
#define	LASSO_SA_STATUS_CODE_ABORT
#define	LASSO_SOAP_BINDING_PROCESS_CONTEXT_PRINCIPAL_OFFLINE
#define	LASSO_SOAP_BINDING_PROCESS_CONTEXT_PRINCIPAL_ONLINE
#define	LASSO_SOAP_BINDING_PROCESS_CONTEXT_SIMULATE

Description

Functions

Types and Values

LASSO_DISCO_HREF

```
#define LASSO_DISCO_HREF            "urn:liberty:disco:2003-08"
```

Namespace for FIXME

LASSO_DISCO_PREFIX

```
#define LASSO_DISCO_PREFIX        "disco"
```

Preferred prefix for namespace of FIXME

LASSO_EP_HREF

```
#define LASSO_EP_HREF   "urn:liberty:id-sis-ep:2003-08"
```

Namespace for FIXME

LASSO_EP_PREFIX

```
#define LASSO_EP_PREFIX "ep"
```

Preferred prefix for namespace of FIXME

LASSO_IS_HREF

```
#define LASSO_IS_HREF "urn:liberty:is:2003-08"
```

Namespace for FIXME

LASSO_IS_PREFIX

```
#define LASSO_IS_PREFIX "is"
```

Preferred prefix for namespace of FIXME

LASSO_PP10_HREF

```
#define LASSO_PP10_HREF   "urn:liberty:id-sis-pp:2003-08"
```

Namespace for ID-SIS Personal Profile

LASSO_PP10_PREFIX

```
#define LASSO_PP10_PREFIX "pp10"
```

Preferred prefix for namespace of ID-SIS Personal Profile

LASSO_SA_HREF

```
#define LASSO_SA_HREF "urn:liberty:sa:2004-04"
```

Namespace for FIXME

LASSO_SA_PREFIX

```
#define LASSO_SA_PREFIX "sa"
```

Preferred prefix for namespace of FIXME

LASSO_SEC_HREF

```
#define LASSO_SEC_HREF "urn:liberty:sec:2003-08"
```

Namespace for FIXME

LASSO_SEC_PREFIX

```
#define LASSO_SEC_PREFIX "sec"
```

Preferred prefix for namespace of FIXME

LASSO_SOAP_BINDING_EXT_HREF

```
#define LASSO_SOAP_BINDING_EXT_HREF "urn:liberty:sb:2004-04"
```

Namespace for FIXME

LASSO_SOAP_BINDING_EXT_PREFIX

```
#define LASSO_SOAP_BINDING_EXT_PREFIX "sbe"
```

Preferred prefix for namespace of FIXME

LASSO_SOAP_BINDING_HREF

```
#define LASSO_SOAP_BINDING_HREF          "urn:liberty:sb:2003-08"
```

Namespace for FIXME

LASSO_SOAP_BINDING_PREFIX

```
#define LASSO_SOAP_BINDING_PREFIX        "sb"
```

Preferred prefix for namespace of FIXME

LASSO_SECURITY_MECH_NULL

```
#define LASSO_SECURITY_MECH_NULL   "urn:liberty:security:2003-08:null:null"
```

LASSO_SECURITY_MECH_X509

```
#define LASSO_SECURITY_MECH_X509   "urn:liberty:security:2005-02:null:X509"
```

LASSO_SECURITY_MECH_SAML

```
#define LASSO_SECURITY_MECH_SAML   "urn:liberty:security:2005-02:null:SAML"
```

LASSO_SECURITY_MECH_BEARER

```
#define LASSO_SECURITY_MECH_BEARER "urn:liberty:security:2005-02:null:Bearer"
```

LASSO_SECURITY_MECH_TLS

```
#define LASSO_SECURITY_MECH_TLS        "urn:liberty:security:2003-08:TLS:null"
```

LASSO_SECURITY_MECH_TLS_X509

```
#define LASSO_SECURITY_MECH_TLS_X509   "urn:liberty:security:2005-02:TLS:X509"
```

LASSO_SECURITY_MECH_TLS_SAML

```
#define LASSO_SECURITY_MECH_TLS_SAML   "urn:liberty:security:2005-02:TLS:SAML"
```

LASSO_SECURITY_MECH_TLS_BEARER

```
#define LASSO_SECURITY_MECH_TLS_BEARER "urn:liberty:security:2005-02:TLS:Bearer"
```

LASSO_SECURITY_MECH_CLIENT_TLS

```
#define LASSO_SECURITY_MECH_CLIENT_TLS        "urn:liberty:security:2003-08:ClientTLS:null"
```

LASSO_SECURITY_MECH_CLIENT_TLS_X509

```
#define LASSO_SECURITY_MECH_CLIENT_TLS_X509   "urn:liberty:security:2005-02:ClientTLS:X509"
```

LASSO_SECURITY_MECH_CLIENT_TLS_SAML

```
#define LASSO_SECURITY_MECH_CLIENT_TLS_SAML   "urn:liberty:security:2005-02:ClientTLS:SAML"
```

LASSO_SECURITY_MECH_CLIENT_TLS_BEARER

```
#define LASSO_SECURITY_MECH_CLIENT_TLS_BEARER "urn:liberty:security:2005-02:ClientTLS: ↩
    Bearer"
```

LASSO_SECURITY11_MECH_X509

```
#define LASSO_SECURITY11_MECH_X509   "urn:liberty:security:2005-02:null:X509"
```

LASSO_SECURITY11_MECH_SAML

```
#define LASSO_SECURITY11_MECH_SAML   "urn:liberty:security:2005-02:null:SAML"
```

LASSO_SECURITY11_MECH_BEARER

```
#define LASSO_SECURITY11_MECH_BEARER "urn:liberty:security:2005-02:null:Bearer"
```

LASSO_SECURITY11_MECH_TLS_X509

```
#define LASSO_SECURITY11_MECH_TLS_X509   "urn:liberty:security:2005-02:TLS:X509"
```

LASSO_SECURITY11_MECH_TLS_SAML

```
#define LASSO_SECURITY11_MECH_TLS_SAML   "urn:liberty:security:2005-02:TLS:SAML"
```

LASSO_SECURITY11_MECH_TLS_BEARER

```
#define LASSO_SECURITY11_MECH_TLS_BEARER "urn:liberty:security:2005-02:TLS:Bearer"
```

LASSO_SA_SASL_SERVICE_NAME

```
#define LASSO_SA_SASL_SERVICE_NAME "idwsf"
```

LASSO_IS_INTERACT_ATTR_INTERACT_IF_NEEDED

```
#define LASSO_IS_INTERACT_ATTR_INTERACT_IF_NEEDED "is:interactIfNeeded"
```

LASSO_IS_INTERACT_ATTR_DO_NOT_INTERACT

```
#define LASSO_IS_INTERACT_ATTR_DO_NOT_INTERACT "is:doNotInteract"
```

LASSO_IS_INTERACT_ATTR_DO_NOT_INTERACT_FOR_DATA

```
#define LASSO_IS_INTERACT_ATTR_DO_NOT_INTERACT_FOR_DATA "is:doNotInteractForData"
```

LASSO_DISCO_STATUS_CODE_OK

```
#define LASSO_DISCO_STATUS_CODE_OK "OK"
```

LASSO_DISCO_STATUS_CODE_DISCO_OK

```
#define LASSO_DISCO_STATUS_CODE_DISCO_OK "disco:OK"
```

LASSO_DISCO_STATUS_CODE_FAILED

```
#define LASSO_DISCO_STATUS_CODE_FAILED "Failed"
```

LASSO_DISCO_STATUS_CODE_REMOVE_ENTRY

```
#define LASSO_DISCO_STATUS_CODE_REMOVE_ENTRY "RemoveEntry"
```

LASSO_DISCO_STATUS_CODE_FORBIDDEN

```
#define LASSO_DISCO_STATUS_CODE_FORBIDDEN "Forbidden"
```

LASSO_DISCO_STATUS_CODE_NO_RESULTS

```
#define LASSO_DISCO_STATUS_CODE_NO_RESULTS "NoResults"
```

LASSO_DISCO_STATUS_CODE_DIRECTIVES

```
#define LASSO_DISCO_STATUS_CODE_DIRECTIVES "Directive"
```

LASSO_DST_STATUS_CODE_OK

```
#define LASSO_DST_STATUS_CODE_OK "OK"
```

LASSO_DST_STATUS_CODE_FAILED

```
#define LASSO_DST_STATUS_CODE_FAILED "Failed"
```

LASSO_DST_STATUS_CODE_PARTIAL

```
#define LASSO_DST_STATUS_CODE_PARTIAL "Partial"
```

LASSO_DST_STATUS_CODE_ACTION_NOT_AUTHORIZED

```
#define LASSO_DST_STATUS_CODE_ACTION_NOT_AUTHORIZED "ActionNotAuthorized"
```

LASSO_DST_STATUS_CODE_ACTION_NOT_SUPPORTED

```
#define LASSO_DST_STATUS_CODE_ACTION_NOT_SUPPORTED "ActionNotSupported"
```

LASSO_DST_STATUS_CODE_ALL_RETURNED

```
#define LASSO_DST_STATUS_CODE_ALL_RETURNED "AllReturned"
```

LASSO_DST_STATUS_CODE_CHANGE_HISTORY_NOT_SUPPORTED

```
#define LASSO_DST_STATUS_CODE_CHANGE_HISTORY_NOT_SUPPORTED "ChangeHistoryNotSupported"
```

LASSO_DST_STATUS_CODE_CHANGED_SINCE_RETURNS_ALL

```
#define LASSO_DST_STATUS_CODE_CHANGED_SINCE_RETURNS_ALL "ChangedSinceReturnsAll"
```

LASSO_DST_STATUS_CODE_DATA_TOO_LONG

```
#define LASSO_DST_STATUS_CODE_DATA_TOO_LONG "DataTooLong"
```

LASSO_DST_STATUS_CODE_EXISTS_ALREADY

```
#define LASSO_DST_STATUS_CODE_EXISTS_ALREADY "ExistsAlready"
```

LASSO_DST_STATUS_CODE_EXTENSION_NOT_SUPPORTED

```
#define LASSO_DST_STATUS_CODE_EXTENSION_NOT_SUPPORTED "ExtensionNotSupported"
```

LASSO_DST_STATUS_CODE_INVALID_DATA

```
#define LASSO_DST_STATUS_CODE_INVALID_DATA "InvalidData"
```

LASSO_DST_STATUS_CODE_INVALID_RESOURCE_ID

```
#define LASSO_DST_STATUS_CODE_INVALID_RESOURCE_ID "InvalidResourceID"
```

LASSO_DST_STATUS_CODE_INVALID_SELECT

```
#define LASSO_DST_STATUS_CODE_INVALID_SELECT "InvalidSelect"
```

LASSO_DST_STATUS_CODE_MISSING_NEW_DATA_ELEMENT

```
#define LASSO_DST_STATUS_CODE_MISSING_NEW_DATA_ELEMENT "MissingNewDataElement"
```

LASSO_DST_STATUS_CODE_MISSING_RESOURCE_ID_ELEMENT

```
#define LASSO_DST_STATUS_CODE_MISSING_RESOURCE_ID_ELEMENT "MissingResourceIDElement"
```

LASSO_DST_STATUS_CODE_MISSING_SELECT

```
#define LASSO_DST_STATUS_CODE_MISSING_SELECT "MissingSelect"
```

LASSO_DST_STATUS_CODE_MODIFIED_SINCE

```
#define LASSO_DST_STATUS_CODE_MODIFIED_SINCE "ModifiedSince"
```

LASSO_DST_STATUS_CODE_NO_MORE_ELEMENTS

```
#define LASSO_DST_STATUS_CODE_NO_MORE_ELEMENTS "NoMoreElements"
```

LASSO_DST_STATUS_CODE_NO_MULTIPLE_ALLOWED

```
#define LASSO_DST_STATUS_CODE_NO_MULTIPLE_ALLOWED "NoMultipleAllowed"
```

LASSO_DST_STATUS_CODE_NO_MULTIPLE_RESOURCES

```
#define LASSO_DST_STATUS_CODE_NO_MULTIPLE_RESOURCES "NoMultipleResources"
```

LASSO_DST_STATUS_CODE_TIME_OUT

```
#define LASSO_DST_STATUS_CODE_TIME_OUT "TimeOut"
```

LASSO_DST_STATUS_CODE_UNEXPECTED_ERROR

```
#define LASSO_DST_STATUS_CODE_UNEXPECTED_ERROR "UnexpectedError"
```

LASSO_SA_STATUS_CODE_OK

```
#define LASSO_SA_STATUS_CODE_OK "OK"
```

LASSO_SA_STATUS_CODE_CONTINUE

```
#define LASSO_SA_STATUS_CODE_CONTINUE "continue"
```

LASSO_SA_STATUS_CODE_ABORT

```
#define LASSO_SA_STATUS_CODE_ABORT "abort"
```

LASSO_SOAP_BINDING_PROCESS_CONTEXT_PRINCIPAL_OFFLINE

```
#define            LASSO_SOAP_BINDING_PROCESS_CONTEXT_PRINCIPAL_OFFLINE
```

LASSO_SOAP_BINDING_PROCESS_CONTEXT_PRINCIPAL_ONLINE

```
#define             LASSO_SOAP_BINDING_PROCESS_CONTEXT_PRINCIPAL_ONLINE
```

LASSO_SOAP_BINDING_PROCESS_CONTEXT_SIMULATE

```
#define             LASSO_SOAP_BINDING_PROCESS_CONTEXT_SIMULATE
```

9.2 LassoDiscoAuthenticateRequester

LassoDiscoAuthenticateRequester — <disco:DirectiveType>

Functions

LassoDiscoAuthenticateRequester *	lasso_disco_authenticate_requester_new ()

Types and Values

struct	LassoDiscoAuthenticateRequester

Description

```
<xs: complexType name="DirectiveType">
 <xs: attribute name="descriptionIDRefs" type="xs:IDREFS" use="optional"/>
</xs: complexType>
<xs: element name="AuthenticateRequester" type="DirectiveType"/>
```

Figure 9.1: Schema fragment for disco:DirectiveType

Functions

lasso_disco_authenticate_requester_new ()

```
LassoDiscoAuthenticateRequester~*
lasso_disco_authenticate_requester_new
                               (void);
```

Types and Values

struct LassoDiscoAuthenticateRequester

```
struct LassoDiscoAuthenticateRequester {
  LassoNode parent;
  gchar *descriptionIDRefs;
};
```

9.3 LassoDiscoAuthenticateSessionContext

LassoDiscoAuthenticateSessionContext — <disco:DirectiveType>

Types and Values

struct	LassoDiscoAuthenticateSessionContext

Description

```
<xs: complexType name="DirectiveType">
 <xs: attribute name="descriptionIDRefs" type="xs:IDREFS" use="optional"/>
</xs: complexType>
<xs: element name="AuthenticateSessionContext" type="DirectiveType"/>
```

Figure 9.2: Schema fragment for disco:DirectiveType

Functions

Types and Values

struct LassoDiscoAuthenticateSessionContext

```
struct LassoDiscoAuthenticateSessionContext {
  LassoNode parent;

  gchar *descriptionIDRefs;
};
```

9.4 LassoDiscoAuthorizeRequester

LassoDiscoAuthorizeRequester — <disco:DirectiveType>

Functions

LassoDiscoAuthorizeRequester *	lasso_disco_authorize_requester_new ()

Types and Values

struct	LassoDiscoAuthorizeRequester

Description

```
<xs: complexType name="DirectiveType">
 <xs: attribute name="descriptionIDRefs" type="xs:IDREFS" use="optional"/>
</xs: complexType>
<xs: element name="AuthorizeRequester" type="DirectiveType"/>
```

Figure 9.3: Schema fragment for disco:DirectiveType

Functions

lasso_disco_authorize_requester_new ()

```
LassoDiscoAuthorizeRequester~*
lasso_disco_authorize_requester_new (void);
```

Types and Values

struct LassoDiscoAuthorizeRequester

```
struct LassoDiscoAuthorizeRequester {
  LassoNode parent;

  gchar *descriptionIDRefs;
};
```

9.5 LassoDiscoCredentials

LassoDiscoCredentials — <disco:Credentials>

Functions

LassoDiscoCredentials *	lasso_disco_credentials_new ()

Types and Values

struct	LassoDiscoCredentials

Description

```
<xs:element name="Credentials" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 9.4: Schema fragment for disco:Credentials

Functions

lasso_disco_credentials_new ()

```
LassoDiscoCredentials~*
lasso_disco_credentials_new (void);
```

Types and Values

struct LassoDiscoCredentials

```
struct LassoDiscoCredentials {
  LassoNode parent;

  GList *any; /* of xmlNode */
};
```

9.6 LassoDiscoDescription

LassoDiscoDescription — <disco:DescriptionType>

Functions

LassoDiscoDescription *	lasso_disco_description_copy ()
LassoDiscoDescription *	lasso_disco_description_new ()
LassoDiscoDescription *	lasso_disco_description_new_with_WsdlRef ()
LassoDiscoDescription *	lasso_disco_description_new_with_BriefSoapHttpDescription ()
gboolean	lasso_disco_description_has_saml_authentication ()
gboolean	lasso_disco_description_has_x509_authentication ()

Types and Values

struct	LassoDiscoDescription

Description

```
<xs:complexType name="DescriptionType">
  <xs:sequence>
    <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="1" maxOccurs="unbounded ↩
        "/>
    <xs:element name="CredentialRef" type="xs:IDREF" minOccurs="0" maxOccurs="unbounded"/>
    <xs:choice>
      <xs:group ref="WsdlRef"/>
      <xs:group ref="BriefSoapHttpDescription"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>

<xs:group name="WsdlRef">
  <xs:sequence>
    <xs:element name="WsdlURI" type="xs:anyURI"/>
    <xs:element name="ServiceNameRef" type="xs:QName"/>
  </xs:sequence>
</xs:group>

<xs:group name="BriefSoapHttpDescription">
  <xs:sequence>
    <xs:element name="Endpoint" type="xs:anyURI"/>
    <xs:element name="SoapAction" type="xs:anyURI" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

Figure 9.5: Schema fragment for disco:DescriptionType

Functions

lasso_disco_description_copy ()

```
LassoDiscoDescription~*
lasso_disco_description_copy (LassoDiscoDescription *description);
```

lasso_disco_description_new ()

```
LassoDiscoDescription~*
lasso_disco_description_new ();
```

lasso_disco_description_new_with_WsdlRef ()

```
LassoDiscoDescription~*
lasso_disco_description_new_with_WsdlRef
                               (const gchar *securityMechID,
                                const gchar *wsdlURI,
                                const gchar *serviceNameRef);
```

lasso_disco_description_new_with_BriefSoapHttpDescription ()

```
LassoDiscoDescription~*
lasso_disco_description_new_with_BriefSoapHttpDescription
                               (const gchar *securityMechID,
                                const gchar *endpoint,
                                const gchar *soapAction);
```

lasso_disco_description_has_saml_authentication ()

```
gboolean
lasso_disco_description_has_saml_authentication
                               (LassoDiscoDescription *description);
```

Checks if the given description supports any security mechanism using SAML authentication.

Parameters

profile	a LassoDiscoDescription

Returns

TRUE if SAML is supported by the service description, FALSE if it is not supported of if description is not a valid LassoDiscoDescription.

lasso_disco_description_has_x509_authentication ()

```
gboolean
lasso_disco_description_has_x509_authentication
                               (LassoDiscoDescription *description);
```

Checks if the given description supports any security mechanism using X509 authentication.

Parameters

profile	a LassoDiscoDescription

Returns

TRUE if X509 is supported by the service description, FALSE if it is not supported of if description is not a valid LassoDiscoDescription.

Types and Values

struct LassoDiscoDescription

```
struct LassoDiscoDescription {
  LassoNode parent;

  /*
   * - The service instance description SHOULD list of all of the security mechanisms that
   *   the service instance supports.
   * - The client SHOULD pick the first mechanism (in the order listed) that it supports;
   *   the description SHOULD list them in order of preference, to avoid situations where the
   *   client fails to gain access to the service because it picked the wrong security
   *   mechanism.
   */
  GList *SecurityMechID; /* of strings */
  GList *CredentialRef; /* of strings */

  /* WsdlRef group */
  gchar *WsdlURI;
  gchar *ServiceNameRef;

  /* BriefSoapHttpDescription group */
  gchar *Endpoint;
  gchar *SoapAction;

  char *id;
};
```

9.7 LassoDiscoEncryptedResourceID

LassoDiscoEncryptedResourceID — <disco:EncryptedResourceID>

Functions

LassoDiscoEncryptedResourceID *	lasso_disco_encrypted_resource_id_new ()

Types and Values

struct	LassoDiscoEncryptedResourceID

Description

```
<xs:element name="EncryptedResourceID" type="EncryptedResourceIDType"/>
<xs:complexType name="EncryptedResourceIDType">
   <xs:sequence>
      <xs:element ref="xenc:EncryptedData"/>
      <xs:element ref="xenc:EncryptedKey"/>
   </xs:sequence>
</xs:complexType>
```

Figure 9.6: Schema fragment for disco:EncryptedResourceID

Functions

lasso_disco_encrypted_resource_id_new ()

```
LassoDiscoEncryptedResourceID~*
lasso_disco_encrypted_resource_id_new ();
```

Types and Values

struct LassoDiscoEncryptedResourceID

```
struct LassoDiscoEncryptedResourceID {
  LassoNode parent;

  xmlNode *EncryptedData;
  GList *EncryptedKey; /* of xmlNode* */
};
```

9.8 LassoDiscoEncryptResourceID

LassoDiscoEncryptResourceID — <disco:DirectiveType>

Functions

LassoDiscoEncryptResourceID *	lasso_disco_encrypt_resource_id_new ()

Types and Values

struct	LassoDiscoEncryptResourceID

Description

```
<xs: complexType name="DirectiveType">
 <xs: attribute name="descriptionIDRefs" type="xs:IDREFS" use="optional"/>
</xs: complexType>
<xs: element name="EncryptResourceID" type="DirectiveType"/>
```

Figure 9.7: Schema fragment for disco:DirectiveType

Functions

lasso_disco_encrypt_resource_id_new ()

```
LassoDiscoEncryptResourceID~*
lasso_disco_encrypt_resource_id_new (void);
```

Types and Values

struct LassoDiscoEncryptResourceID

```
struct LassoDiscoEncryptResourceID {
  LassoNode parent;

  gchar *descriptionIDRefs;
};
```

9.9 LassoDiscoGenerateBearerToken

LassoDiscoGenerateBearerToken — <disco:DirectiveType>

Functions

LassoDiscoGenerateBearerToken *	lasso_disco_generate_bearer_token_new ()

Types and Values

struct	LassoDiscoGenerateBearerToken

Description

```
<xs: complexType name="DirectiveType">
 <xs: attribute name="descriptionIDRefs" type="xs:IDREFS" use="optional"/>
</xs: complexType>
<xs: element name="GenerateBearerToken" type="disco: DirectiveType"/>
```

Figure 9.8: Schema fragment for disco:DirectiveType

Functions

lasso_disco_generate_bearer_token_new ()

```
LassoDiscoGenerateBearerToken~*
lasso_disco_generate_bearer_token_new (void);
```

Types and Values

struct LassoDiscoGenerateBearerToken

```
struct LassoDiscoGenerateBearerToken {
  LassoNode parent;
  gchar *descriptionIDRefs;
};
```

9.10 LassoDiscoInsertEntry

LassoDiscoInsertEntry — <disco:InsertEntryType>

Functions

LassoDiscoInsertEntry *	lasso_disco_insert_entry_new ()

Types and Values

struct	LassoDiscoInsertEntry

Description

```
<xs:complexType name="InsertEntryType">
  <xs:sequence>
    <xs:element ref="ResourceOffering"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Figure 9.9: Schema fragment for disco:InsertEntryType

Functions

lasso_disco_insert_entry_new ()

```
LassoDiscoInsertEntry~*
lasso_disco_insert_entry_new (LassoDiscoResourceOffering *resourceOffering);
```

Types and Values

struct LassoDiscoInsertEntry

```
struct LassoDiscoInsertEntry {
  LassoNode parent;

  LassoDiscoResourceOffering *ResourceOffering;
  GList *any; /* of LassoNode */
};
```

9.11 LassoDiscoModifyResponse

LassoDiscoModifyResponse — <disco:ModifyResponse>

Functions

LassoDiscoModifyResponse *	lasso_disco_modify_response_new ()

Types and Values

struct	LassoDiscoModifyResponse

Description

```
<xs:element name="ModifyResponse" type="ModifyResponseType"/>
<xs:complexType name="ModifyResponseType">
  <xs:sequence>
    <xs:element ref="Status"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="newEntryIDs" use="optional">
    <xs:simpleType>
      <xs:list itemType="IDReferenceType"/>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

Schema fragment (liberty-idwsf-utility-1.0-errata-v1.0.xsd):

<xs:simpleType name="IDReferenceType">
  <xs:annotation>
    <xs:documentation> This type can be used when referring to elements that are
      identified using an IDType </xs:documentation>
    </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.10: Schema fragment for disco:ModifyResponse

Functions

lasso_disco_modify_response_new ()

```
LassoDiscoModifyResponse~*
lasso_disco_modify_response_new (LassoUtilityStatus *status);
```

Types and Values

struct LassoDiscoModifyResponse

```
struct LassoDiscoModifyResponse {
  LassoNode parent;

  /*
   * - QNames define in strings.h (LASSO_DISCO_STATUS_CODE_* ) are expected to appear in
   *   the"code" attribute of Status elements used in Discovery Service protocol messages.
   * - The "ref" attribute on the Status element is not used in this specification,
   *   so it MUST not appear on Status elements in Discovery Service protocol messages.
   * - The contents of the "comment" attribute are not defined by this specification,
   *   but it may be used for additional descriptive text intended for human consumption
   *   (for example, to carry information that will aid debugging).
  */
  LassoUtilityStatus *Status;

  char *id;
  char *newEntryIDs;
};
```

9.12 LassoDiscoModify

LassoDiscoModify — <disco:Modify>

Functions

LassoDiscoModify *	lasso_disco_modify_new ()

Types and Values

struct	LassoDiscoModify

Description

```
<xs:element name="Modify" type="ModifyType"/>
<xs:complexType name="ModifyType">
  <xs:sequence>
    <xs:group ref="ResourceIDGroup"/>
    <xs:element name="InsertEntry" type="InsertEntryType" minOccurs="0" maxOccurs=" ↩
        unbounded"/>
    <xs:element name="RemoveEntry" type="RemoveEntryType" minOccurs="0" maxOccurs=" ↩
        unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:group name="ResourceIDGroup">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="1">
      <xs:element ref="ResourceID"/>
      <xs:element ref="EncryptedResourceID"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
```

Figure 9.11: Schema fragment for disco:Modify

Functions

lasso_disco_modify_new ()

```
LassoDiscoModify~*
lasso_disco_modify_new (void);
```

Types and Values

struct LassoDiscoModify

```
struct LassoDiscoModify {
  LassoNode parent;

  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;

  GList *InsertEntry; /* of LassoNode */
  GList *RemoveEntry; /* of LassoNode */

  char *id;
};
```

9.13 LassoDiscoOptions

LassoDiscoOptions — <disco:Options>

Functions

LassoDiscoOptions *	lasso_disco_options_new ()

Types and Values

struct	LassoDiscoOptions

Description

```
<xs:complexType name="OptionsType">
  <xs:sequence>
    <xs:element ref="Option" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Figure 9.12: Schema fragment for disco:Options

Functions

lasso_disco_options_new ()

```
LassoDiscoOptions~*
lasso_disco_options_new (void);
```

Types and Values

struct LassoDiscoOptions

```
struct LassoDiscoOptions {
  LassoNode parent;

  GList *Option; /* of strings */
};
```

9.14 LassoDiscoQueryResponse

LassoDiscoQueryResponse — <disco:QueryResponse>

Functions

LassoDiscoQueryResponse *	lasso_disco_query_response_new ()

Types and Values

struct	LassoDiscoQueryResponse

Description

```
<xs:complexType name="QueryResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status"/>

    <xs:element ref="wsa:EndpointReference"
      minOccurs="0"
      maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

Figure 9.13: Schema fragment for disco:QueryResponse

Functions

lasso_disco_query_response_new ()

```
LassoDiscoQueryResponse~*
lasso_disco_query_response_new (LassoUtilityStatus *status);
```

Types and Values

struct LassoDiscoQueryResponse

```
struct LassoDiscoQueryResponse {
  LassoNode parent;

  LassoUtilityStatus *Status;
  GList *ResourceOffering; /* of LassoNode */
  LassoDiscoCredentials *Credentials;

  char *id;
};
```

9.15 LassoDiscoQuery

LassoDiscoQuery — <disco:Query>

Functions

LassoDiscoQuery *	lasso_disco_query_new ()

Types and Values

struct	LassoDiscoQuery

Description

```
<xs:complexType name="QueryType">
  <xs:sequence>
    <xs:element name="RequestedService"
      type="RequestedServiceType"
      minOccurs="0"
      maxOccurs="unbounded"/>
    </xs:sequence>

    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

Figure 9.14: Schema fragment for disco:Query

Functions

lasso_disco_query_new ()

```
LassoDiscoQuery~*
lasso_disco_query_new (void);
```

Types and Values

struct LassoDiscoQuery

```
struct LassoDiscoQuery {
  LassoNode parent;

  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;
  GList *RequestedServiceType; /* of LassoNode */
  gchar *id;
};
```

9.16 LassoDiscoRemoveEntry

LassoDiscoRemoveEntry — <disco:RemoveEntryType>

Functions

LassoDiscoRemoveEntry *	lasso_disco_remove_entry_new ()

Types and Values

struct	LassoDiscoRemoveEntry

Description

```
<xs:complexType name="RemoveEntryType">
  <xs:attribute name="entryID" type="IDReferenceType" use="required"/>
</xs:complexType>

Schema fragment (liberty-idwsf-utility-1.0-errata-v1.0.xsd)

<xs:simpleType name="IDReferenceType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.15: Schema fragment for disco:RemoveEntryType

Functions

lasso_disco_remove_entry_new ()

```
LassoDiscoRemoveEntry~*
lasso_disco_remove_entry_new (const gchar *entryID);
```

Types and Values

struct LassoDiscoRemoveEntry

```
struct LassoDiscoRemoveEntry {
  LassoNode parent;

  char *entryID;
};
```

9.17 LassoDiscoRequestedServiceType

LassoDiscoRequestedServiceType — <disco:RequestedServiceType>

Functions

LassoDiscoRequestedServiceType *	lasso_disco_requested_service_type_new ()

Types and Values

struct	LassoDiscoRequestedServiceType

Description

```
<xs:element name="RequestedServiceType" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
     <xs:sequence>
       <xs:element ref="ServiceType"/>
       <xs:element ref="Options" minOccurs="0"/>
     </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ServiceType" type="xs:anyURI"/>
```

Figure 9.16: Schema fragment for disco:RequestedServiceType

Functions

lasso_disco_requested_service_type_new ()

```
LassoDiscoRequestedServiceType~*
lasso_disco_requested_service_type_new
                                (const char *serviceType);
```

Types and Values

struct LassoDiscoRequestedServiceType

```
struct LassoDiscoRequestedServiceType {
  LassoNode parent;

  char  *ServiceType;
  LassoDiscoOptions *Options;
};
```

9.18 LassoDiscoResourceID

LassoDiscoResourceID — <disco:ResourceID>

Functions

LassoDiscoResourceID *	lasso_disco_resource_id_new ()

Types and Values

struct	LassoDiscoResourceID

Description

```
<xs:element name="ResourceID" type="ResourceIDType"/>
<xs:complexType name="ResourceIDType">
    <xs:simpleContent>
       <xs:extension base="xs:anyURI">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
       </xs:extension>
    </xs:simpleContent>
 </xs:complexType>
```

Figure 9.17: Schema fragment for disco:ResourceID

Functions

lasso_disco_resource_id_new ()

```
LassoDiscoResourceID~*
lasso_disco_resource_id_new (const gchar *content);
```

Types and Values

struct LassoDiscoResourceID

```
struct LassoDiscoResourceID {
  LassoNode parent;

  gchar *content;

  gchar *id;
};
```

9.19 LassoDiscoResourceOffering

LassoDiscoResourceOffering — <disco:ResourceOffering>

Functions

LassoDiscoResourceOffering *	lasso_disco_resource_offering_new ()

Types and Values

struct	LassoDiscoResourceOffering

Description

```
<xs:element name="ResourceOffering" type="ResourceOfferingType"/>
<xs:complexType name="ResourceOfferingType">
  <xs:sequence>
    <xs:group ref="ResourceIDGroup"/>
    <xs:element name="ServiceInstance" type="ServiceInstanceType"/>
    <xs:element ref="Options" minOccurs="0"/>
    <xs:element name="Abstract" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="entryID" type="IDType" use="optional"/>
</xs:complexType>

Schema fragment (liberty-idwsf-utility-1.0-errata-v1.0.xsd)

<xs:simpleType name="IDType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.18: Schema fragment for disco:ResourceOffering

Functions

lasso_disco_resource_offering_new ()

```
LassoDiscoResourceOffering~*
lasso_disco_resource_offering_new (LassoDiscoServiceInstance *serviceInstance);
```

Types and Values

struct LassoDiscoResourceOffering

```
struct LassoDiscoResourceOffering {
  LassoNode parent;

  /* elements */
  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;
  LassoDiscoServiceInstance *ServiceInstance;

  /*
   * If the Options element is present, but it is empty, it means that the service instance
   * explicitly advertises that none of the options are available.
  */
  LassoDiscoOptions *Options;
  gchar *Abstract;

  /* attributes */
  gchar *entryID;
};
```

9.20 LassoDiscoSendSingleLogout

LassoDiscoSendSingleLogout — <disco:DirectiveType>

Functions

LassoDiscoSendSingleLogout *	lasso_disco_send_single_logout_new ()

Types and Values

struct	LassoDiscoSendSingleLogout

Description

```
<xs: complexType name="DirectiveType">
 <xs: attribute name="descriptionIDRefs" type="xs:IDREFS" use="optional"/>
</xs: complexType>
<xs: element name="SendSingleLogOut" type="disco: DirectiveType"/>
```

Figure 9.19: Schema fragment for disco:DirectiveType

Functions

lasso_disco_send_single_logout_new ()

```
LassoDiscoSendSingleLogout~*
lasso_disco_send_single_logout_new (void);
```

Types and Values

struct LassoDiscoSendSingleLogout

```
struct LassoDiscoSendSingleLogout {
  LassoNode parent;

  gchar *descriptionIDRefs;
};
```

9.21 LassoDiscoServiceInstance

LassoDiscoServiceInstance — <disco:ServiceInstanceType>

Functions

LassoDiscoServiceInstance *	lasso_disco_service_instance_copy ()
LassoDiscoServiceInstance *	lasso_disco_service_instance_new ()

Types and Values

struct	LassoDiscoServiceInstance

Description

```
<xs:complexType name="ServiceInstanceType">
  <xs:sequence>
    <xs:element ref="ServiceType"/>
    <xs:element name="ProviderID" type="md:entityIDType"/>
    <xs:element name="Description" type="DescriptionType" minOccurs="1" maxOccurs=" ↩
        unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="ServiceType" type="xs:anyURI"/>
```

Figure 9.20: Schema fragment for disco:ServiceInstanceType

Functions

lasso_disco_service_instance_copy ()

```
LassoDiscoServiceInstance~*
lasso_disco_service_instance_copy (LassoDiscoServiceInstance *serviceInstance);
```

lasso_disco_service_instance_new ()

```
LassoDiscoServiceInstance~*
lasso_disco_service_instance_new (const gchar *serviceType,
                                  const gchar *providerID,
                                  LassoDiscoDescription *description);
```

Types and Values

struct LassoDiscoServiceInstance

```
struct LassoDiscoServiceInstance {
  LassoNode parent;

  char *ServiceType;
  char *ProviderID;
  GList *Description; /* of LassoNode */
};
```

9.22 LassoDstData

LassoDstData — <dst:Data>

Functions

LassoDstData *	lasso_dst_data_new ()

Types and Values

struct	LassoDstData

Description

```
<xs:element name="Data" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute name="itemIDRef" type="IDReferenceType"/>
  </xs:complexType>
</xs:element>

Schema fragment (liberty-idwsf-utility-1.0-errata-v1.0.xsd)

<xs:simpleType name="IDReferenceType">
  <xs:annotation>
    <xs:documentation> This type can be used when referring to elements that are
      identified using an IDType </xs:documentation>
    </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.21: Schema fragment for dst:Data

Functions

lasso_dst_data_new ()

```
LassoDstData~*
lasso_dst_data_new (void);
```

Types and Values

struct LassoDstData

```
struct LassoDstData {
  LassoNode parent;

  GList *any; /* of xmlNode* */

  char *id;
  char *itemIDRef;
};
```

9.23 LassoDstModification

LassoDstModification — <dst:Modification>

Functions

LassoDstModification *	lasso_dst_modification_new ()

Types and Values

struct	LassoDstModification

Description

```

<xs:element name="Modification" maxOccurs="unbounded">
<xs:complexType>
    <xs:sequence>
        <xs:element name="Select" type="SelectType"/>
        <xs:element name="NewData" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:any minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute name="notChangedSince" type="xs:dateTime"/>
    <xs:attribute name="overrideAllowed" type="xs:boolean" default="0"/>
</xs:complexType>
</xs:element>
```

Figure 9.22: Schema fragment for dst:Modification

Functions

lasso_dst_modification_new ()

```
LassoDstModification~*
lasso_dst_modification_new (const char *select);
```

Types and Values

struct LassoDstModification

```
struct LassoDstModification {
  LassoNode parent;

  char *Select;
  LassoDstNewData *NewData;

  char *id;
  char *notChangedSince;
  gboolean overrideAllowed;
};
```

9.24 LassoDstModifyResponse

LassoDstModifyResponse — <dst:ModifyResponse>

Functions

LassoDstModifyResponse *	lasso_dst_modify_response_new ()

Types and Values

struct	LassoDstModifyResponse

Description

```
<xs:element name="ModifyResponse" type="ResponseType"/>
<xs:complexType name="ResponseType">
    <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute name="itemIDRef" type="IDReferenceType"/>
    <xs:attribute name="timeStamp" type="xs:dateTime"/>
</xs:complexType>
```

Figure 9.23: Schema fragment for dst:ModifyResponse

Functions

lasso_dst_modify_response_new ()

```
LassoDstModifyResponse~*
lasso_dst_modify_response_new (LassoUtilityStatus *status);
```

Types and Values

struct LassoDstModifyResponse

```
struct LassoDstModifyResponse {
  LassoNode parent;

  GList *Extension; /* of xmlNode* */
  LassoUtilityStatus *Status;

  char *id;
  char *itemIDRef;
  char *timeStamp;
};
```

9.25 LassoDstModify

LassoDstModify — <dst:Modify>

Functions

LassoDstModify *	lasso_dst_modify_new ()

Types and Values

struct	LassoDstModify

Description

```
<xs:element name="Modify" type="ModifyType"/>
<xs:complexType name="ModifyType">
    <xs:sequence>
        <xs:group ref="ResourceIDGroup" minOccurs="0"/>
        <xs:element name="Modification" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Select" type="SelectType"/>
                    <xs:element name="NewData" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:any minOccurs="0" maxOccurs="unbounded"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="id" type="xs:ID"/>
                <xs:attribute name="notChangedSince" type="xs:dateTime"/>
                <xs:attribute name="overrideAllowed" type="xs:boolean" default="0"/>
            </xs:complexType>
        </xs:element>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute name="itemID" type="IDType"/>
</xs:complexType>
```

Figure 9.24: Schema fragment for dst:Modify

Functions

lasso_dst_modify_new ()

```
LassoDstModify~*
lasso_dst_modify_new ();
```

Types and Values

struct LassoDstModify

```
struct LassoDstModify {
  LassoNode parent;

  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;
  GList *Modification; /* of LassoNode */
  GList *Extension; /* of xmlNode* */

  char *id;
  char *itemID;
};
```

9.26 LassoDstNewData

LassoDstNewData — <dst:NewData>

Functions

LassoDstNewData *	lasso_dst_new_data_new ()

Types and Values

struct	LassoDstNewData

Description

```
<xs:element name="NewData" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Figure 9.25: Schema fragment for dst:NewData

Functions

lasso_dst_new_data_new ()

```
LassoDstNewData~*
lasso_dst_new_data_new (void);
```

Types and Values

struct LassoDstNewData

```
struct LassoDstNewData {
  LassoNode parent;

  GList *any; /* of xmlNode* */
};
```

9.27 LassoDstQueryItem

LassoDstQueryItem — <dst:QueryItem>

Functions

LassoDstQueryItem *	lasso_dst_query_item_new ()

Types and Values

struct	LassoDstQueryItem

Description

```
<xs:element name="QueryItem" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Select" type="SelectType"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute name="includeCommonAttributes" type="xs:boolean" default="0"/>
    <xs:attribute name="itemID" type="IDType"/>
    <xs:attribute name="changedSince" type="xs:dateTime"/>
  </xs:complexType>
</xs:element>

Schema fragment (liberty-idwsf-utility-1.0-errata-v1.0.xsd):

<xs:simpleType name="IDType">
  <xs:annotation>
    <xs:documentation>
      This type should be used to provided IDs to components that have IDs
      that may not  be scoped within the local xml instance document.
    </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.26: Schema fragment for dst:QueryItem

Functions

lasso_dst_query_item_new ()

```
LassoDstQueryItem~*
lasso_dst_query_item_new (const char *select,
                          const char *item_id);
```

Types and Values

struct LassoDstQueryItem

```
struct LassoDstQueryItem {
  LassoNode parent;

  char *Select;
  char *id;
  gboolean includeCommonAttributes;
  char *itemID;
  char *changedSince;
};
```

9.28 LassoDstQueryResponse

LassoDstQueryResponse — <dst:QueryResponse>

Functions

LassoDstQueryResponse *	lasso_dst_query_response_new ()

Types and Values

struct	LassoDstQueryResponse

Description

```
<xs:element name="QueryResponse" type="QueryResponseType"/>
<xs:complexType name="QueryResponseType">
  <xs:sequence>
    <xs:element ref="Status"/>
    <xs:element name="Data" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID"/>
  <xs:attribute name="itemIDRef" type="IDReferenceType"/>
  <xs:attribute name="timeStamp" type="xs:dateTime"/>
</xs:complexType>

Schema fragment (liberty-idwsf-utility-1.0-errata-v1.0.xsd):

<xs:simpleType name="IDReferenceType">
  <xs:annotation>
    <xs:documentation> This type can be used when referring to elements that are
      identified using an IDType </xs:documentation>
    </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.27: Schema fragment for dst:QueryResponse

Functions

lasso_dst_query_response_new ()

```
LassoDstQueryResponse~*
lasso_dst_query_response_new (LassoUtilityStatus *status);
```

Types and Values

struct LassoDstQueryResponse

```
struct LassoDstQueryResponse {
  LassoNode parent;

  LassoUtilityStatus *Status;
  GList *Data; /* of LassoNode */
  GList *Extension; /* of xmlNode* */

  char *id;
  char *itemIDRef;
  char *timeStamp;
};
```

9.29 LassoDstQuery

LassoDstQuery — <dst:Query>

Functions

LassoDstQuery *	lasso_dst_query_new ()

Types and Values

struct	LassoDstQuery

Description

```
<xs:element name="Query" type="QueryType"/>
<xs:complexType name="QueryType">
    <xs:sequence>
        <xs:group ref="ResourceIDGroup" minOccurs="0"/>
   <xs:element name="QueryItem" maxOccurs="unbounded"/>
   <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute name="itemID" type="IDType"/>
</xs:complexType>

<xs:simpleType name="IDReferenceType">
  <xs:annotation>
    <xs:documentation> This type can be used when referring to elements that are
      identified using an IDType </xs:documentation>
    </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

Figure 9.28: Schema fragment for dst:Query

Functions

lasso_dst_query_new ()

```
LassoDstQuery~*
lasso_dst_query_new (LassoDstQueryItem *query_item);
```

Creates a new LassoDstQuery object. If *query_item* is set it is added to the requested query items.

Parameters

query_item	query item to embed in request (optional)

Returns

a newly created LassoDstQuery object.

Types and Values

struct LassoDstQuery

```
struct LassoDstQuery {
  LassoNode parent;

  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;
  GList *QueryItem; /* of LassoNode */
  GList *Extension; /* of xmlNode* */
  char *id;
  char *itemID;
};
```

9.30 LassoIsRedirectRequest

LassoIsRedirectRequest — <is:RedirectRequest>

Functions

#define	LASSO_IS_IS_REDIRECT_REQUEST()
LassoIsRedirectRequest *	lasso_is_redirect_request_new ()

Types and Values

struct	LassoIsRedirectRequest

Description

```
<xs:element name="RedirectRequest" type="RedirectRequestType"/>
<xs:complexType name="RedirectRequestType">
   <xs:attribute name="redirectURL" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Figure 9.29: Schema fragment for is:RedirectRequest

Functions

LASSO_IS_IS_REDIRECT_REQUEST()

```
#define              LASSO_IS_IS_REDIRECT_REQUEST(obj)
```

lasso_is_redirect_request_new ()

```
LassoIsRedirectRequest~*
lasso_is_redirect_request_new (const char *redirectURL);
```

Types and Values

struct LassoIsRedirectRequest

```
struct LassoIsRedirectRequest {
  LassoNode parent;

  char *redirectURL;
};
```

9.31 LassoIsUserInteraction

LassoIsUserInteraction — <is:UserInteraction>

Functions

#define	LASSO_IS_IS_USER_INTERACTION()
LassoIsUserInteraction *	lasso_is_user_interaction_new ()

Types and Values

struct	LassoIsUserInteraction

Description

```
<xs:element name="UserInteraction" type="UserInteractionHeaderType"/>
<xs:complexType name="UserInteractionHeaderType">
  <xs:sequence>
    <xs:element name="InteractionService" type="disco:ResourceOfferingType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="interact" type="xs:QName" use="optional" default="is:interactIfNeeded ↩
      "/>
  <xs:attribute name="language" type="xs:NMTOKENS" use="optional"/>
  <xs:attribute name="redirect" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="maxInteractTime" type="xs:integer" use="optional"/>
  <xs:attribute ref="soap:actor" use="optional"/>
  <xs:attribute ref="soap:mustUnderstand" use="optional"/>
</xs:complexType>
```

Figure 9.30: Schema fragment for is:UserInteraction

Functions

LASSO_IS_IS_USER_INTERACTION()

```
#define             LASSO_IS_IS_USER_INTERACTION(obj)
```

lasso_is_user_interaction_new ()

```
LassoIsUserInteraction~*
lasso_is_user_interaction_new (void);
```

Types and Values

struct LassoIsUserInteraction

```
struct LassoIsUserInteraction {
  LassoNode parent;

  GList *InteractionService; /* of LassoNode */

  gchar *id;
  gchar *interact;
  gchar *language;
  gboolean redirect;
  gint maxInteractTime;

  /* FIXME : implement soap:actor and soap:mustUnderstand */
};
```

9.32 LassoSaCredentials

LassoSaCredentials — <sa:Credentials>

Functions

LassoSaCredentials *	lasso_sa_credentials_new ()

Types and Values

struct	LassoSaCredentials

Description

```
    <xs:element name="Credentials" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded ↩
            "/>
      </xs:sequence>
    </xs:complexType>
    </xs:element>
```

Figure 9.31: Schema fragment for sa:Credentials

Functions

lasso_sa_credentials_new ()

```
LassoSaCredentials~*
lasso_sa_credentials_new ();
```

Types and Values

struct LassoSaCredentials

```
struct LassoSaCredentials {
  LassoNode parent;

  GList *any; /* of LassoNode */
};
```

9.33 LassoSaParameter

LassoSaParameter — <sa:Parameter>

Functions

LassoSaParameter *	lasso_sa_parameter_new ()

Types and Values

struct	LassoSaParameter

Description

```
<xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:element>
```

Figure 9.32: Schema fragment for sa:Parameter

Functions

lasso_sa_parameter_new ()

```
LassoSaParameter~*
lasso_sa_parameter_new ();
```

Types and Values

struct LassoSaParameter

```
struct LassoSaParameter {
  LassoNode parent;

  char *content;

  char *name;
};
```

9.34 LassoSaPasswordTransforms

LassoSaPasswordTransforms — <sa:PasswordTransforms>

Functions

LassoSaPasswordTransforms *	lasso_sa_password_transforms_new ()

Types and Values

struct	LassoSaPasswordTransforms

Description

```
  <xs:element name="PasswordTransforms">
     <xs:annotation>
       <xs:documentation>
         Contains ordered list of sequential password transformations
       </xs:documentation>
     </xs:annotation>
     <xs:complexType>
       <xs:sequence>
         <xs:element name="Transform" maxOccurs="unbounded">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded">
               <xs:complexType>
                 <xs:simpleContent>
                   <xs:extension base="xs:string">
                     <xs:attribute name="name" type="xs:string" use="required"/>
                   </xs:extension>
                 </xs:simpleContent>
               </xs:complexType>
               </xs:eledment>
             </xs:sequence>
             <xs:attribute name="name" type="xs:anyURI" use="required"/>
             <xs:attribute name="id" type="xs:ID"use="optional"/>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
 </xs:element>
```

Figure 9.33: Schema fragment for sa:PasswordTransforms

Functions

lasso_sa_password_transforms_new ()

```
LassoSaPasswordTransforms~*
lasso_sa_password_transforms_new ();
```

Types and Values

struct LassoSaPasswordTransforms

```
struct LassoSaPasswordTransforms {
  LassoNode parent;

  GList *Transform; /* of LassoNode */
};
```

9.35 LassoSaSASLRequest

LassoSaSASLRequest — <sa:SASLRequest>

Functions

LassoSaSASLRequest *	lasso_sa_sasl_request_new ()

Types and Values

struct	LassoSaSASLRequest

Description

```
<xs:element name="SASLRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Data" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary"/>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element ref="lib:RequestAuthnContext" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="mechanism"type="xs:string" use="required"/>
    <xs:attribute name="authzID" type="xs:string" use="optional"/>
    <xs:attribute name="advisoryAuthnID" type="xs:string" use="optional"/>
    <xs:attribute name="id" type="xs:ID"use="optional"/>
  </xs:complexType>
</xs:element>
```

Figure 9.34: Schema fragment for sa:SASLRequest

Functions

lasso_sa_sasl_request_new ()

```
LassoSaSASLRequest~*
lasso_sa_sasl_request_new (const char *mechanism);
```

Types and Values

struct LassoSaSASLRequest

```
struct LassoSaSASLRequest {
  LassoNode parent;

  GList *Data; /* of string */
  LassoLibRequestAuthnContext *RequestAuthnContext;

  char *mechanism;
  char *authzID;
  char *advisoryAuthnID;
  char *id;
};
```

9.36 LassoSaSASLResponse

LassoSaSASLResponse — <sa:SASLResponse>

Functions

LassoSaSASLResponse *	lasso_sa_sasl_response_new ()

Types and Values

struct	LassoSaSASLResponse

Description

```
<xs:element name="SASLResponse">
<xs:complexType>
  <xs:sequence>
    <xs:element ref="Status"/>
    <xs:element ref="PasswordTransforms" minOccurs="0"/>
    <xs:element name="Data" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:base64Binary"/>
      </xs:simpleContent>
    </xs:complexType>
    </xs:element>
    <xs:element ref="disco:ResourceOffering" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Credentials" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded ↩
            "/>
      </xs:sequence>
    </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="serverMechanism" type="xs:string" ="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>
```

Figure 9.35: Schema fragment for sa:SASLResponse

Functions

lasso_sa_sasl_response_new ()

```
LassoSaSASLResponse~*
lasso_sa_sasl_response_new (LassoUtilityStatus *status);
```

Types and Values

struct LassoSaSASLResponse

```
struct LassoSaSASLResponse {
  LassoNode parent;

  LassoUtilityStatus *Status;
  GList *PasswordTransforms; /* of LassoNode */
  GList *Data; /* of strings */
  GList *ResourceOffering; /* of LassoNode */
  GList *Credentials; /* of LassoNode */
  GList *any; /* of LassoNode */

  gchar *serverMechanism;
  gchar *id;
};
```

9.37 LassoSaTransform

LassoSaTransform — <sa:Transform>

Functions

LassoSaTransform *	lasso_sa_transform_new ()

Types and Values

struct	LassoSaTransform

Description

```
<xs:element name="Transform" maxOccurs="unbounded">
<xs:complexType>
   <xs:sequence>
     <xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded">
       <xs:complexType>
         <xs:simpleContent>
           <xs:extension base="xs:string">
             <xs:attribute name="name" type="xs:string" use="required"/>
           </xs:extension>
         </xs:simpleContent>
       </xs:complexType>
     </xs:element>
   </xs:sequence>
   <xs:attribute name="name" type="xs:anyURI" use="required"/>
   <xs:attribute name="id" type="xs:ID"use="optional"/>
 </xs:complexType>
 </xs:element>
```

Figure 9.36: Schema fragment for sa:Transform

Functions

lasso_sa_transform_new ()

```
LassoSaTransform~*
lasso_sa_transform_new ();
```

Types and Values

struct LassoSaTransform

```
struct LassoSaTransform {
  LassoNode parent;

  GList *Parameter; /* of LassoNode */

  char *name;
  char *id;
};
```

9.38 LassoSecResourceAccessStatement

LassoSecResourceAccessStatement —

Functions

LassoNode *	lasso_sec_resource_access_statement_new ()

Types and Values

struct	LassoSecResourceAccessStatement

Description

Functions

lasso_sec_resource_access_statement_new ()

```
LassoNode~*
lasso_sec_resource_access_statement_new
                               (void);
```

Creates a new LassoSecResourceAccessStatement object.

Returns

a newly created LassoSecResourceAccessStatement object

Types and Values

struct LassoSecResourceAccessStatement

```
struct LassoSecResourceAccessStatement {
  LassoSamlSubjectStatementAbstract parent;
};
```

9.39 LassoSoapBindingConsent

LassoSoapBindingConsent — <soapbinding:ConsentType>

Functions

LassoSoapBindingConsent *	lasso_soap_binding_consent_new ()

Types and Values

struct	LassoSoapBindingConsent

Description

```
<xs:complexType name="ConsentType">
  <xs:attribute name="uri" type="xs:anyURI" use="required"/>
  <xs:attribute name="timestamp" type="xs:dateTime" use="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
<xs:element name="Consent" type="ConsentType"/>
```

Figure 9.37: Schema fragment for soapbinding:ConsentType

Functions

lasso_soap_binding_consent_new ()

```
LassoSoapBindingConsent~*
lasso_soap_binding_consent_new (const gchar *uri);
```

Types and Values

struct LassoSoapBindingConsent

```
struct LassoSoapBindingConsent {
  LassoNode parent;

  gchar *uri;
  gchar *timestamp;
  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.40 LassoSoapBindingCorrelation

LassoSoapBindingCorrelation — <soapbinding:correlationType>

Functions

LassoSoapBindingCorrelation *	lasso_soap_binding_correlation_new ()

Types and Values

struct	LassoSoapBindingCorrelation

Description

```
<xs:complexType name="correlationType">
  <xs:attribute name="messageID" type="IDType" use="required"/>
  <xs:attribute name="refToMessageID" type="IDType" use="optional"/>
  <xs:attribute name="timestamp" type="xs: dateTime" use="required"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

Figure 9.38: Schema fragment for soapbinding:correlationType

Functions

lasso_soap_binding_correlation_new ()

```
LassoSoapBindingCorrelation~*
lasso_soap_binding_correlation_new (const gchar *messageId,
                                    const gchar *timestamp);
```

Types and Values

struct LassoSoapBindingCorrelation

```
struct LassoSoapBindingCorrelation {
  LassoNode parent;

  gchar *messageID;
  gchar *refToMessageID;
  gchar *timestamp;
  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.41 LassoSoapBindingExtCredentialsContext

LassoSoapBindingExtCredentialsContext — <soapbinding:CredentialsContextType>

Types and Values

struct	LassoSoapBindingExtCredentialsContext

Description

```
<xs:complexType name="CredentialsContextType">
  <xs:sequence>
    <xs:element ref="lib:RequestAuthnContext" minOccurs="0"/>
    <xs:element name="SecurityMechID" type="xs: anyURI" minOccurs="0" maxOccurs="unbounded ↩
        "/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
<xs:element name="CredentialsContext" type="CredentialsContextType"/>
```

Figure 9.39: Schema fragment for soapbinding:CredentialsContextType

Functions

Types and Values

struct LassoSoapBindingExtCredentialsContext

```
struct LassoSoapBindingExtCredentialsContext {
  LassoNode parent;

  LassoLibRequestAuthnContext *RequestAuthnContext;
  gchar *SecurityMechID;

  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.42 LassoSoapBindingExtCredential

LassoSoapBindingExtCredential — <soapbinding:Credential>

Types and Values

struct	LassoSoapBindingExtCredential

Description

```
<xs:element name="Credential" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
     <xs:sequence>
        <xs:any namespace="##any" processContents="lax"/>
     </xs:sequence>
     <xs:attribute name="notOnOrAfter" type="xs:dateTime" use="optional"/>
  </xs:complexType>
</xs:element>
```

Figure 9.40: Schema fragment for soapbinding:Credential

Functions

Types and Values

struct LassoSoapBindingExtCredential

```
struct LassoSoapBindingExtCredential {
  LassoNode parent;

  GList *any; /* of LassoNode */

  gchar *notOnOrAfter;
};
```

9.43 LassoSoapBindingExtServiceInstanceUpdate

LassoSoapBindingExtServiceInstanceUpdate — <soapbinding:ServiceInstanceUpdateType>

Types and Values

struct	LassoSoapBindingExtServiceInstanceUpdate

Description

```
<xs:complexType name="ServiceInstanceUpdateType">
  <xs:sequence>
      <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded ↩
          "/>
      <xs:element name="Credential" minOccurs="0" maxOccurs="unbounded">
         <xs:complexType>
            <xs:sequence>
               <xs:any namespace="##any" processContents="lax"/>
            </xs:sequence>
            <xs:attribute name="notOnOrAfter" type="xs:dateTime" use="optional"/>
         </xs:complexType>
      </xs:element>
      <xs:element name="Endpoint" type="xs:anyURI" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

Figure 9.41: Schema fragment for soapbinding:ServiceInstanceUpdateType

Functions

Types and Values

struct LassoSoapBindingExtServiceInstanceUpdate

```
struct LassoSoapBindingExtServiceInstanceUpdate {
  LassoNode parent;

  gchar *SecurityMechID;
  LassoSoapBindingExtCredential *Credential;
  gchar *Endpoint;

  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.44 LassoSoapBindingExtTimeout

LassoSoapBindingExtTimeout — <soapbinding:TimeoutType>

Functions

LassoSoapBindingExtTimeout *	lasso_soap_binding_ext_timeout_new ()

Types and Values

struct	LassoSoapBindingExtTimeout

Description

```
<xs:complexType name="TimeoutType">
  <xs:attribute name="maxProcessingTime" type="xs:integer" use="required"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
<xs:element name="Timeout" type="TimeoutType"/>
```

Figure 9.42: Schema fragment for soapbinding:TimeoutType

Functions

lasso_soap_binding_ext_timeout_new ()

```
LassoSoapBindingExtTimeout~*
lasso_soap_binding_ext_timeout_new (gint maxProcessingTime);
```

Types and Values

struct LassoSoapBindingExtTimeout

```
struct LassoSoapBindingExtTimeout {
  LassoNode parent;

  gint maxProcessingTime;
  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.45 LassoSoapBindingProcessingContext

LassoSoapBindingProcessingContext — <soapbinding:ProcessingContextType>

Functions

LassoSoapBindingProcessingContext *	lasso_soap_binding_processing_context_new ()

Types and Values

struct	LassoSoapBindingProcessingContext

Description

```
<xs:complexType name="ProcessingContextType">
  <xs:simpleContent>
      <xs:extension base="xs:anyURI">
         <xs:attribute name="id" type="xs:ID" use="optional"/>
         <xs:attribute ref="S:mustUnderstand" use="optional"/>
         <xs:attribute ref="S:actor" use="optional"/>
      </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="ProcessingContext" type="ProcessingContextType"/>
```

Figure 9.43: Schema fragment for soapbinding:ProcessingContextType

Functions

lasso_soap_binding_processing_context_new ()

```
LassoSoapBindingProcessingContext~*
lasso_soap_binding_processing_context_new
                               ();
```

Types and Values

struct LassoSoapBindingProcessingContext

```
struct LassoSoapBindingProcessingContext {
  LassoNode parent;

  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
  gchar *content;
};
```

9.46 LassoSoapBindingProvider

LassoSoapBindingProvider — <soapbinding:ProviderType>

Functions

LassoSoapBindingProvider *	lasso_soap_binding_provider_new ()

Types and Values

struct	LassoSoapBindingProvider

Description

```
<xs:complexType name="ProviderType">
  <xs:attribute name="providerID" type="xs:anyURI" use="required"/>
  <xs:attribute name="affiliationID" type="xs: nyURI" use="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
<xs:element name="Provider" type="ProviderType"/>
```

Figure 9.44: Schema fragment for soapbinding:ProviderType

Functions

lasso_soap_binding_provider_new ()

```
LassoSoapBindingProvider~*
lasso_soap_binding_provider_new (const gchar *providerID);
```

Types and Values

struct LassoSoapBindingProvider

```
struct LassoSoapBindingProvider {
  LassoNode parent;

  gchar *providerID;
  gchar *affiliationID;

  gchar *id;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.47 LassoSoapBindingUsageDirective

LassoSoapBindingUsageDirective — <soapbinding:UsageDirectiveType>

Functions

LassoSoapBindingUsageDirective *	lasso_soap_binding_usage_directive_new ()

Types and Values

struct	LassoSoapBindingUsageDirective

Description

```
<xs:complexType name="UsageDirectiveType">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
    <xs:attribute name="ref" type="xs:IDREF" use="required"/>
    <xs:attribute ref="S:mustUnderstand" use="optional"/>
    <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
<xs:element name="UsageDirective" type="UsageDirectiveType"/>
```

Figure 9.45: Schema fragment for soapbinding:UsageDirectiveType

Functions

lasso_soap_binding_usage_directive_new ()

```
LassoSoapBindingUsageDirective~*
lasso_soap_binding_usage_directive_new
                               (const gchar *ref);
```

Types and Values

struct LassoSoapBindingUsageDirective

```
struct LassoSoapBindingUsageDirective {
  LassoNode parent;

  GList *other; /* of LassoNode */

  gchar *id;
  gchar *ref;
  gchar *mustUnderstand;
  gchar *actor;
};
```

9.48 LassoUtilityStatus

LassoUtilityStatus — <utility:Status>

Functions

LassoUtilityStatus *	lasso_utility_status_new ()

Types and Values

struct	LassoUtilityStatus

Description

```
<xs:element name="Status" type="StatusType">
  <xs:annotation>
    <xs:documentation> A standard Status type</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="StatusType">
  <xs:annotation>
    <xs:documentation> A type that may be used for status codes. </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="Status" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="code" type="xs:QName" use="required"/>
  <xs:attribute name="ref" type="xs:NCName" use="optional"/>
  <xs:attribute name="comment" type="xs:string" use="optional"/>
</xs:complexType>
```

Figure 9.46: Schema fragment for utility:Status

Functions

lasso_utility_status_new ()

```
LassoUtilityStatus~*
lasso_utility_status_new (const char *code);
```

Types and Values

struct LassoUtilityStatus

```
struct LassoUtilityStatus {
  LassoNode parent;

  LassoUtilityStatus *Status;
  char *code;
  char *ref;
  char *comment;
};
```

Chapter 10

ID-WSF 2.0

10.1 SOAP Binding utility function for ID-WSF 2.0

SOAP Binding utility function for ID-WSF 2.0 —

Functions

char *	lasso_soap_envelope_sb2_get_provider_id ()
const char *	lasso_soap_envelope_sb2_get_redirect_request_url ()
LassoIdWsf2Sb2TargetIdentity *	lasso_soap_envelope_sb2_get_target_identity_header ()
LassoIdWsf2Sb2UserInteractionHeader *	lasso_soap_envelope_get_sb2_user_interaction_header ()

Description

Functions

lasso_soap_envelope_sb2_get_provider_id ()

```
char~*
lasso_soap_envelope_sb2_get_provider_id
                              (LassoSoapEnvelope *soap_envelope);
```

Return the providerID contained in the sb2:Sender header.

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

the providerID string or NULL if no sb2:Sender header is present.

lasso_soap_envelope_sb2_get_redirect_request_url ()

```
const char~*
```

```
lasso_soap_envelope_sb2_get_redirect_request_url
                               (LassoSoapEnvelope *soap_envelope);
```

Return the redirect request URL from the sb2:RedirectRequest SOAP Fault detail.

The WSC MUST verify that this URL belong to the WSP. You can do this by comparing the domain with the one in the *LassoProfile.msg_url* field

The WSC MUST redirect the User Agent to this URL with a GET or POST request. It MUST add a parameter named ReturnToURL giving the URL where the WSP will send the User Agent after the interaction. It MAY add an IDP parameter indicating to the WSP how to authenticate the principal if no preexisting session with the User Agent exists

The WSP must check that the ReturnToURL belon to the WSP, by using the providerID URL for example.

After the interaction the WSC must redirect the User Agent to the ReturnToURL URL adding a parameter named ResendMessage. If ResendMessage is 0 or false, it means that the principal refused to continue the process. Any other value means that the prinicpal accepted and so the WSC can try again its request.

In order to succeed the request need to refer to the SOAP Fault response containing the RedirectRequest element. See lasso_soap_envelop and LassoWsAddrAttributedURI.

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

the redirect URL string or NULL if no sb2:RedirectRequest detail is present.

[transfer none][allow-none]

lasso_soap_envelope_sb2_get_target_identity_header ()

```
LassoIdWsf2Sb2TargetIdentity~*
lasso_soap_envelope_sb2_get_target_identity_header
                               (LassoSoapEnvelope *soap_envelope);
```

Return the first sb2:TargetIdentity header.

Parameters

soap_envelope	a LassoSoapEnvelope object.

Returns

the first LassoIdWsf2Sb2TargetIdentity object found in the headers of the *soap_envelope* , or NULL if none is found.

[transfer none]

lasso_soap_envelope_get_sb2_user_interaction_header ()

```
LassoIdWsf2Sb2UserInteractionHeader~*
lasso_soap_envelope_get_sb2_user_interaction_header
                               (LassoSoapEnvelope *soap_envelope,
                                gboolean create);
```

10.2 LassoIdWsf2Discovery

LassoIdWsf2Discovery —

Functions

LassoIdWsf2Discovery *	lasso_idwsf2_discovery_new ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_register ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_association_add ()
lasso_error_t	lasso_idwsf2_discovery_init_query ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_query ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_replace ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_delete ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_association_delete ()
lasso_error_t	lasso_idwsf2_discovery_init_metadata_association_query ()
lasso_error_t	lasso_idwsf2_discovery_add_service_metadata ()
lasso_error_t	lasso_idwsf2_discovery_add_simple_service_metadata ()
GList *	lasso_idwsf2_discovery_get_metadatas ()
lasso_error_t	lasso_idwsf2_discovery_add_requested_service ()
lasso_error_t	lasso_idwsf2_discovery_build_request_msg ()
lasso_error_t	lasso_idwsf2_discovery_process_request_msg ()
LassoIdWsf2DiscoveryRequestType	lasso_idwsf2_discovery_get_request_type ()
lasso_error_t	lasso_idwsf2_discovery_validate_request ()
lasso_error_t	lasso_idwsf2_discovery_fail_request ()
lasso_error_t	lasso_idwsf2_discovery_process_response_msg ()
GList *	lasso_idwsf2_discovery_get_endpoint_references ()
GList *	lasso_idwsf2_discovery_get_svcmdids ()
void	lasso_idwsf2_discovery_set_svcmdids ()

Types and Values

struct	LassoIdWsf2Discovery

Description

Functions

lasso_idwsf2_discovery_new ()

```
LassoIdWsf2Discovery~*
lasso_idwsf2_discovery_new (LassoServer *server);
```

Create a new LassoIdWsf2Discovery.

Parameters

server	a LassoServer object, for resolving ProviderID names.	*[allow-none]*

Returns

a newly created LassoIdWsf2Discovery object; or NULL if an error occured.

lasso_idwsf2_discovery_init_metadata_register ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_register
                                        (LassoIdWsf2Discovery *discovery);
```

Initialise a ID-WSF service metadata registration request to a Discovery service.

Parameters

discovery	a LassoIdWsf2Discovery object

Returns

0 on success; an error code otherwise.

lasso_idwsf2_discovery_init_metadata_association_add ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_association_add
                                        (LassoIdWsf2Discovery *discovery);
```

Initialise a request to associate a user account to a service metadata, allowing a WSC to request this service for data related to this user account.

Parameters

discovery	a LassoIdWsf2Discovery
svcMDID	identifier of the service metadata the user wants to associate with

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_discovery_init_query ()

```
lasso_error_t
lasso_idwsf2_discovery_init_query (LassoIdWsf2Discovery *discovery);
```

Initialise a request for ID-WSF discovery Query to a discovery service.

Parameters

discovery	a LassoIdWsf2Discovery

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_discovery_init_metadata_query ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_query
                              (LassoIdWsf2Discovery *discovery);
```

Initialise a ID-WSF service metadata query request to a Discovery service

Parameters

discovery	a LassoIdWsf2Discovery object

Returns

0 on success; an error code otherwise.

lasso_idwsf2_discovery_init_metadata_replace ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_replace
                              (LassoIdWsf2Discovery *discovery);
```

Initialise a ID-WSF service metadata replace request to a Discovery service.

Parameters

discovery	a LassoIdWsf2Discovery object

Returns

0 on success; an error code otherwise.

lasso_idwsf2_discovery_init_metadata_delete ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_delete
                              (LassoIdWsf2Discovery *discovery);
```

Initialise a ID-WSF service metadata query request to a Discovery service

Parameters

discovery	a LassoIdWsf2Discovery object

Returns

0 on success; an error code otherwise.

lasso_idwsf2_discovery_init_metadata_association_delete ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_association_delete
                               (LassoIdWsf2Discovery *discovery);
```

lasso_idwsf2_discovery_init_metadata_association_query ()

```
lasso_error_t
lasso_idwsf2_discovery_init_metadata_association_query
                               (LassoIdWsf2Discovery *discovery);
```

lasso_idwsf2_discovery_add_service_metadata ()

```
lasso_error_t
lasso_idwsf2_discovery_add_service_metadata
                               (LassoIdWsf2Discovery *idwsf2_discovery,
                                LassoIdWsf2DiscoSvcMetadata *service_metadata);
```

Add a new metadata object to a request.

Parameters

idwsf2_discovery	a LassoIdWsf2Discovery object
service_metadata	a LassoIdWsf2DiscoSvcMetadata object to add to the register request.

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_discovery_add_simple_service_metadata ()

```
lasso_error_t
lasso_idwsf2_discovery_add_simple_service_metadata
                               (LassoIdWsf2Discovery *idwsf2_discovery,
                                const char *abstract,
                                const char *provider_id,
                                GList *service_types,
                                GList *options,
                                const char *address,
                                GList *security_mechanisms);
```

Add new metadata to the current Metadata Register request.

Parameters

idwsf2_discovery	a LassoIdWsf2Discovery object	
abstract	a human description of the service.	*[allow-none]*
provider_id	the provider id of the service to register, if none is given, providerId of the current LassoServer object is used.	*[allow-none]*
service_types	an array of service type URIs.	*[element-type utf8][allow-none]*
options	an array of option string.	*[element-type LassoIdWsf2DiscoOptions][allow-none]*
address	the URI of the service endpoint for the default EndpointContext.	*[allow-none]*
security_mechanisms	the security mechanisms supported by the service.	*[allow-none][element-type utf8]*

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_discovery_get_metadatas ()

```
GList~*
lasso_idwsf2_discovery_get_metadatas (LassoIdWsf2Discovery *discovery);
```

Return the current list of metadatas in the `discovery` object. They can be metadatas just received through a LASSO_IDWSF2_DISCOV request or added through lasso_idwsf2_discovery_add_service_metadata() or lasso_idwsf2_discovery_add_simple_service_metadata().

Parameters

discovery	a LassoIdWsf2Discovery object	

Returns

the list of metadatas.

[transfer none][element-type LassoIdWsf2DiscoSvcMetadata]

lasso_idwsf2_discovery_add_requested_service ()

```
lasso_error_t
lasso_idwsf2_discovery_add_requested_service
                               (LassoIdWsf2Discovery *discovery,
                                GList *service_types,
                                GList *provider_ids,
                                GList *options,
                                GList *security_mechanisms,
                                GList *frameworks,
                                GList *actions,
                                LassoIdWsf2DiscoveryQueryResultType result_type,
                                const char *req_id);
```

Add a new request to find some specific services associated to the current principal at the discovery service.

Parameters

discovery	a LassoIdWsf2Discovery	
service_types	the service type (or data profile) requested.	*[element-type utf8][allow-none]*
provider_ids	the providers ids to select.	*[element-type utf8][allow-none]*
options	the options to select.	*[element-type utf8][allow-none]*
security_mechanisms	the security mechanisms to select.	*[element-type utf8][allow-none]*
frameworks	the ID-WSF framework version to select.	*[element-type utf8][allow-none]*
actions	the actions to select.	*[element-type utf8][allow-none]*
result_type	how to filter the generated EPRs.	*[allow-none][default LASSO_IDWSF2_DISCOVERY_QUERY_RESULT_TYPE_N*
req_id	an eventual ID to put on the request, that can be matched with the generated EndpointReferences.	*[allow-none]*

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_discovery_build_request_msg ()

```
lasso_error_t
lasso_idwsf2_discovery_build_request_msg
                               (LassoIdWsf2Discovery *discovery,
                                const char *security_mechanism);
```

Build the request message using a security mechanism to authenticate the requester and the target identity. If none is given Bearer mechanism is used.

Parameters

discovery	a LassoIdWsf2Discovery object	
security_mechanism	the security mech id to use, if NULL a Bearer mechanism is used.	*[allow-none]*

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_discovery_process_request_msg ()

```
lasso_error_t
lasso_idwsf2_discovery_process_request_msg
                               (LassoIdWsf2Discovery *discovery,
                                const char *message);
```

Parse a Discovery service request.

Parameters

discovery	a LassoIdWsf2Discovery object	
message	a received SOAP message	

Returns

0 if sucessful, an error code otherwise among:

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if *profile* is not a LassoIdWsf2Profile object,
- LASSO_PARAM_ERROR_INVALID_VALUE if message is NULL,
- LASSO_PROFILE_ERROR_INVALID_MSG if we cannot parse the message,
- LASSO_SOAP_ERROR_MISSING_BODY if the message has no body content.

lasso_idwsf2_discovery_get_request_type ()

```
LassoIdWsf2DiscoveryRequestType
lasso_idwsf2_discovery_get_request_type
                               (LassoIdWsf2Discovery *discovery);
```

Return the type of the last parsed request.

Parameters

discovery	a LassoIdWsf2Discovery object	

Returns

the type of the last parsed request.

lasso_idwsf2_discovery_validate_request ()

```
lasso_error_t
lasso_idwsf2_discovery_validate_request
                               (LassoIdWsf2Discovery *discovery);
```

Accept the discovery request, and produce the response.

Parameters

discovery	a LassoIdWsf2Discovery object	

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_discovery_fail_request ()

```
lasso_error_t
lasso_idwsf2_discovery_fail_request (LassoIdWsf2Discovery *discovery,
                                      const char *status_code,
                                      const char *status_code2);
```

Fail the last request with the given status code.

Parameters

discovery	a LassoIdWsf2Discovery	
status_code	a status code string	
status_code2	a second-level status code.	*[allow-none]*

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_discovery_process_response_msg ()

```
lasso_error_t
lasso_idwsf2_discovery_process_response_msg
                                (LassoIdWsf2Discovery *discovery,
                                 const char *msg);
```

Parameters

discovery	a LassoIdWsf2Discovery object	
msg	a string containing the response messages	

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_discovery_get_endpoint_references ()

```
GList~*
lasso_idwsf2_discovery_get_endpoint_references
                                (LassoIdWsf2Discovery *discovery);
```

Return the list of wsa:EndpointReference returned by the last discovery query.

Parameters

discovery	a LassoIdWsf2Discovery object	

Returns

a GList of LassoWsAddrEndpointReference objects, or NULL if none is found.

[transfer none][element-type LassoWsAddrEndpointReference]

lasso_idwsf2_discovery_get_svcmdids ()

```
GList~*
lasso_idwsf2_discovery_get_svcmdids (LassoIdWsf2Discovery *discovery);
```

Return the list of SvcMDID, or service metadata ids, returned by the last discovery query.

Parameters

discovery	a LassoIdWsf2Discovery object	

Returns

a list of SvcMDID's.

[transfer none][element-type utf8][allow-none]

lasso_idwsf2_discovery_set_svcmdids ()

```
void
lasso_idwsf2_discovery_set_svcmdids (LassoIdWsf2Discovery *discovery,
                                     GList *svcmdids);
```

Set the list of SvcMDID, or service metadata ids.

Parameters

discovery	a LassoIdWsf2Discovery object	
svcmdids	a list of service metadata IDs.	*[element-type utf8][allow-none]*

Types and Values

struct LassoIdWsf2Discovery

```
struct LassoIdWsf2Discovery {
  LassoIdWsf2Profile parent;
};
```

10.3 LassoIdWsf2DataService

LassoIdWsf2DataService —

Functions

LassoIdWsf2DataService *	lasso_idwsf2_data_service_new ()
lasso_error_t	lasso_idwsf2_data_service_init_create ()
lasso_error_t	lasso_idwsf2_data_service_init_delete ()
lasso_error_t	lasso_idwsf2_data_service_init_modify ()
lasso_error_t	lasso_idwsf2_data_service_init_query ()
lasso_error_t	lasso_idwsf2_data_service_add_modify_item ()
lasso_error_t	lasso_idwsf2_data_service_add_namespace ()
lasso_error_t	lasso_idwsf2_data_service_add_query_item ()
lasso_error_t	lasso_idwsf2_data_service_build_request_msg ()
lasso_error_t	lasso_idwsf2_data_service_build_response_msg ()
LassoNode *	lasso_idwsf2_data_service_get_item ()
LassoIdWsf2DstRefData *	lasso_idwsf2_data_service_get_query_item_result ()
GList *	lasso_idwsf2_data_service_get_query_item_results ()
LassoIdWsf2DataServiceRequestType	lasso_idwsf2_data_service_get_request_type ()
const char *	lasso_idwsf2_data_service_get_service_type ()
const char *	lasso_idwsf2_data_service_get_service_type_prefix ()
lasso_error_t	lasso_idwsf2_data_service_process_request_msg ()
lasso_error_t	lasso_idwsf2_data_service_process_response_msg ()
lasso_error_t	lasso_idwsf2_data_service_set_query_item_result ()
char *	lasso_idwsf2_data_service_get_query_item_result_content ()
lasso_error_t	lasso_idwsf2_data_service_set_service_type ()
lasso_error_t	lasso_idwsf2_data_service_set_status_code ()
GList *	lasso_idwsf2_data_service_get_item_ids ()
GList *	lasso_idwsf2_data_service_get_items ()
LassoIdWsf2UtilStatus *	lasso_idwsf2_data_service_get_response_status ()
lasso_error_t	lasso_idwsf2_data_service_validate_request ()

Types and Values

struct	LassoIdWsf2DataService

Description

Functions

lasso_idwsf2_data_service_new ()

```
LassoIdWsf2DataService~*
lasso_idwsf2_data_service_new (LassoServer *server);
```

Create a new LassoIdWsf2DataService.

Parameters

server	a LassoServer object, for resolving ProviderIDs.	*[allow-none]*

Returns

a newly created LassoIdWsf2DataService object

lasso_idwsf2_data_service_init_create ()

```
lasso_error_t
lasso_idwsf2_data_service_init_create (LassoIdWsf2DataService *service);
```

lasso_idwsf2_data_service_init_delete ()

```
lasso_error_t
lasso_idwsf2_data_service_init_delete (LassoIdWsf2DataService *service);
```

lasso_idwsf2_data_service_init_modify ()

```
lasso_error_t
lasso_idwsf2_data_service_init_modify (LassoIdWsf2DataService *service);
```

Initialise an ID-WSF 2.0 DataService modify request.

Parameters

service	a LassoIdWsf2DataService

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_data_service_init_query ()

```
lasso_error_t
lasso_idwsf2_data_service_init_query (LassoIdWsf2DataService *service);
```

Initialise an ID-WSF 2.0 DataService query request.

Parameters

service	a LassoIdWsf2DataService

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_data_service_add_modify_item ()

```
lasso_error_t
lasso_idwsf2_data_service_add_modify_item
                              (LassoIdWsf2DataService *service,
                               const gchar *item_query,
                               xmlNode *new_data,
                               gboolean overrideAllowed,
                               const gchar *item_id);
```

Add an item in the modification request.

Parameters

service	a LassoIdWsf2DataService	
item_query	XPATH of the item to modify	
new_data	new value for the selected item.	*[allow-none]*
overrideAllowed	FALSE means only allowing to create a new item, but not modify existing one, TRUE means allowing to modify existing item.	*[allow-none][default FALSE]*
item_id	identifier of the item to modify.	*[allow-none]*

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_data_service_add_namespace ()

```
lasso_error_t
lasso_idwsf2_data_service_add_namespace
                              (LassoIdWsf2DataService *data_service,
                               const char *prefix,
                               const char *href);
```

Add a new namespace to use for example in XPath elements or in Data or NewData objects.

Parameters

service	a LassoIdWsf2DataService object

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_data_service_add_query_item ()

```
lasso_error_t
lasso_idwsf2_data_service_add_query_item
```

```
                                (LassoIdWsf2DataService *service,
                                 const gchar *item_query,
                                 const gchar *item_id);
```

Add an item in the query request.

Parameters

service	a LassoIdWsf2DataService	
item_query	a query string	
item_id	identifier of the queried item, which will allow to retrieve it in the response.	*[allow-none]*

Returns

0 on success; or a negative value otherwise.

lasso_idwsf2_data_service_build_request_msg ()

```
lasso_error_t
lasso_idwsf2_data_service_build_request_msg
                                (LassoIdWsf2DataService *service,
                                 const char *security_mech_id);
```

Build the request message.

Parameters

service	a LassoIdWsf2DataService object	
security_mech_id	the security mechanism to employ, default is Bearer mechanism.	*[allow-none]*

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_data_service_build_response_msg ()

```
lasso_error_t
lasso_idwsf2_data_service_build_response_msg
                                (LassoIdWsf2DataService *service);
```

Build the response message corresponding to the current request.

Parameters

service	a LassoIdWsf2DataService object

Returns

0 if successfull, an error code otherwise.

lasso_idwsf2_data_service_get_item ()

```
LassoNode~*
lasso_idwsf2_data_service_get_item (LassoIdWsf2DataService *data_service,
                                    const char *item_id);
```

Retrieve a specific item from a request.

Parameters

service	a LassoIdWsf2DataService object	
item_id	the itemID of the item to return, if NULL try to get the only one item (if there is more than one, it returns NULL).	

Returns

a LassoIdWsf2DstRefQueryItem or a LassoIdWsf2DstRefModifyItem object, or NULL if no item for the given item_id exists.

[transfer none][allow-none]

lasso_idwsf2_data_service_get_query_item_result ()

```
LassoIdWsf2DstRefData~*
lasso_idwsf2_data_service_get_query_item_result
                               (LassoIdWsf2DataService *service,
                                const char *item_id);
```

Parameters

service	a LassoIdWsf2DataService object	
item_id	an item_id or NULL if only one data is present.	*[allow-none]*

Returns

a LassoIdWsf2DstRefData or NULL if none is found.

[allow-none][transfer none]

lasso_idwsf2_data_service_get_query_item_results ()

```
GList~*
lasso_idwsf2_data_service_get_query_item_results
                               (LassoIdWsf2DataService *service);
```

Parameters

service	a LassoIdWsf2DataService object

Returns

the list of LassoIdWsf2DstRefData or NULL if none is found.

[allow-none][transfer none][element-type LassoIdWsf2DstRefData]

lasso_idwsf2_data_service_get_request_type ()

```
LassoIdWsf2DataServiceRequestType
lasso_idwsf2_data_service_get_request_type
                               (LassoIdWsf2DataService *service);
```

Return the type of the currently handled request.

Parameters

service	a LassoIdWsf2DataService object

lasso_idwsf2_data_service_get_service_type ()

```
const char~*
lasso_idwsf2_data_service_get_service_type
                               (LassoIdWsf2DataService *service);
```

Return the service type of the received request

Parameters

service	a LassoIdWsf2DataService object

Returns

the URI of the service type or NULL.

[allow-none][transfer none]

lasso_idwsf2_data_service_get_service_type_prefix ()

```
const char~*
lasso_idwsf2_data_service_get_service_type_prefix
                               (LassoIdWsf2DataService *service);
```

Return the service type prefix of the received request

Parameters

service	a LassoIdWsf2DataService object	

Returns

the URI of the service type prefix or NULL.

[allow-none][transfer none]

lasso_idwsf2_data_service_process_request_msg ()

```
lasso_error_t
lasso_idwsf2_data_service_process_request_msg
                               (LassoIdWsf2DataService *service,
                                const char *msg);
```

Process a newly received requests.

Parameters

service	a LassoIdWsf2DataService object	
msg	the message string	

lasso_idwsf2_data_service_process_response_msg ()

```
lasso_error_t
lasso_idwsf2_data_service_process_response_msg
                               (LassoIdWsf2DataService *service,
                                const char *msg);
```

Process a received SOAP message response.

Parameters

service	a LassoIdWsf2DataService object	
msg	the message content.	*[allow-none]*

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_data_service_set_query_item_result ()

```
lasso_error_t
lasso_idwsf2_data_service_set_query_item_result
                               (LassoIdWsf2DataService *data_service,
                                const char *item_id,
                                xmlNode *xml_data,
                                gboolean add);
```

Set result data for a certain query-item.

Parameters

service	a LassoIdWsf2DataService object	
item_id	target a certain QueryItem if NULL, means there is only one query item.	*[allow-none]*
xml_data	the data to add.	*[allow-none]*
add	add data to existing datas.	*[allow-none][default FALSE]*

lasso_idwsf2_data_service_get_query_item_result_content ()

```
char~*
lasso_idwsf2_data_service_get_query_item_result_content
                              (LassoIdWsf2DataService *service,
                               const char *item_id);
```

Returns the text content of the query item result identified by `item_id` or the only query item result if `item_id` is NULL.

If `item_id` is NULL and there is multiple results, returns NULL.

Parameters

service	a LassoIdWsf2DataService object	
item_id	the identifier of the result asked, if NULL and there is only one respone, returns it.	*[allow-none]*

Returns

the text content of the query item result.

[transfer full]

lasso_idwsf2_data_service_set_service_type ()

```
lasso_error_t
lasso_idwsf2_data_service_set_service_type
                              (LassoIdWsf2DataService *service,
                               const char *prefix,
                               const char *service_type);
```

Fix a service type for this `service` .

Parameters

service	a LassoIdWsf2DataService object	
prefix	a prefix to use in producing XML documents	
service_type	the service type URI	

lasso_idwsf2_data_service_set_status_code ()

```
lasso_error_t
lasso_idwsf2_data_service_set_status_code
                               (LassoIdWsf2DataService *service,
                                const char *status_code,
                                const char *status_code2);
```

Set the status code for the current response, if no response exists, it starts one using lasso_idwsf2_data_service_validate_request(), if it fails, report a SOAP Fault.

Parameters

service	a LassoIdWsf2DataService	
status_code	a first level status code	
status_code2	a second level status code	

lasso_idwsf2_data_service_get_item_ids ()

```
GList~*
lasso_idwsf2_data_service_get_item_ids
                               (LassoIdWsf2DataService *data_service);
```

Return the list of items ids for the currently handled request.

Parameters

service	a LassoIdWsf2DataService object	

Returns

a list of string ids, or NULL if none is found. The caller must free the return value.

[element-type utf8][transfer full]

lasso_idwsf2_data_service_get_items ()

```
GList~*
lasso_idwsf2_data_service_get_items (LassoIdWsf2DataService *data_service);
```

Parameters

service	a LassoIdWsf2DataService object	

Returns

a list of Query or Modify items, or NULL if none is found.

[element-type LassoNode][transfer none]

lasso_idwsf2_data_service_get_response_status ()

```
LassoIdWsf2UtilStatus~*
lasso_idwsf2_data_service_get_response_status
                               (LassoIdWsf2DataService *service);
```

Return the status from the current response.

Parameters

service	a LassoIdWsf2UtilStatus object

Returns

a LassoIdWsf2UtilStatus object, or NULL.

[transfer none][allow-none]

lasso_idwsf2_data_service_validate_request ()

```
lasso_error_t
lasso_idwsf2_data_service_validate_request
                               (LassoIdWsf2DataService *service);
```

Initialize a new response object corresponding to the current request. If not request if found or the request is invalid, a failure response is created.

Parameters

service	a LassoIdWsf2DataService object

Returns

0 if successful, or LASSO_PROFILE_ERROR_INVALID_REQUEST.

Types and Values

struct LassoIdWsf2DataService

```
struct LassoIdWsf2DataService {
  LassoIdWsf2Profile parent;
};
```

10.4 LassoIdWsf2Profile

LassoIdWsf2Profile —

Functions

void	lasso_idwsf2_profile_set_epr ()
LassoWsAddrEndpointReference *	lasso_idwsf2_profile_get_epr ()
LassoSoapEnvelope *	lasso_idwsf2_profile_get_soap_envelope_request ()
LassoSoapEnvelope *	lasso_idwsf2_profile_get_soap_envelope_response ()
lasso_error_t	lasso_idwsf2_profile_init_request ()
lasso_error_t	lasso_idwsf2_profile_build_request_msg ()
lasso_error_t	lasso_idwsf2_profile_process_request_msg ()
lasso_error_t	lasso_idwsf2_profile_check_security_mechanism ()
LassoNode *	lasso_idwsf2_profile_get_name_identifier ()
lasso_error_t	lasso_idwsf2_profile_init_response ()
lasso_error_t	lasso_idwsf2_profile_init_soap_fault_response ()
lasso_error_t	lasso_idwsf2_profile_redirect_user_for_interaction ()
lasso_error_t	lasso_idwsf2_profile_build_response_msg ()
lasso_error_t	lasso_idwsf2_profile_process_response_msg ()

Types and Values

struct	LassoIdWsf2Profile

Description

Functions

lasso_idwsf2_profile_set_epr ()

```
void
lasso_idwsf2_profile_set_epr (LassoIdWsf2Profile *idwsf2_profile,
                              LassoWsAddrEndpointReference *epr);
```

lasso_idwsf2_profile_get_epr ()

```
LassoWsAddrEndpointReference~*
lasso_idwsf2_profile_get_epr (LassoIdWsf2Profile *idwsf2_profile);
```

Return the EPR used by this profile.

Parameters

idwsf2_profile	a LassoIdWsf2Profile object

Returns

a LassoWsAddrEndpointReference object, or NULL if none is set.

[transfer none]

lasso_idwsf2_profile_get_soap_envelope_request ()

```
LassoSoapEnvelope~*
lasso_idwsf2_profile_get_soap_envelope_request
                               (LassoIdWsf2Profile *idwsf2_profile);
```

Return the last parsed SOAP request object.

Parameters

idwsf2_profile	a LassoIdWsf2Profile object

Returns

a LassoSoapEnvelope object or NULL if no request as ever been parsed with this object. You must free this object.

[transfer none]

lasso_idwsf2_profile_get_soap_envelope_response ()

```
LassoSoapEnvelope~*
lasso_idwsf2_profile_get_soap_envelope_response
                              (LassoIdWsf2Profile *idwsf2_profile);
```

Return the last parsed SOAP response object.

Parameters

idwsf2_profile	a LassoIdWsf2Profile object

Returns

a LassoSoapEnvelope object or NULL if no response as ever been parsed with this objects. You must free this object.

[transfer none]

lasso_idwsf2_profile_init_request ()

```
lasso_error_t
lasso_idwsf2_profile_init_request (LassoIdWsf2Profile *profile);
```

Initialize a new SOAP ID-WSF 2.0 request. Clear the existing request if one is currently set.

Parameters

profile	a LassoIdWsf2Profile object

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_profile_build_request_msg ()

```
lasso_error_t
lasso_idwsf2_profile_build_request_msg
```

```
                              (LassoIdWsf2Profile *profile,
                               const char *security_mech_id);
```

Serialize and sign, if needed, the SOAP request message, put the result in

```
LASSO_PROFILE(profile)->msg_body
```

.

FIXME: really do sign messages.

Parameters

profile	a LassoIdWsf2Profile object

Returns

0 if successful, LASSO_PROFILE_ERROR_BUILDING_REQUEST_FAILED.

lasso_idwsf2_profile_process_request_msg ()

```
lasso_error_t
lasso_idwsf2_profile_process_request_msg
                              (LassoIdWsf2Profile *profile,
                               const char *msg);
```

Parse a SOAP request message and initialize the SOAP Envelope for the response.

Parameters

wsf2_profile	a LassoIdWsf2Profile object
message	a received SOAP message

Returns

0 if successful, an error code otherwise among:

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if *profile* is not a LassoIdWsf2Profile object,
- LASSO_PARAM_ERROR_INVALID_VALUE if message is NULL,
- LASSO_PROFILE_ERROR_INVALID_MSG if we cannot parse the message,
- LASSO_SOAP_ERROR_MISSING_BODY if the message has no body content.

lasso_idwsf2_profile_check_security_mechanism ()

```
lasso_error_t
lasso_idwsf2_profile_check_security_mechanism
                              (LassoIdWsf2Profile *profile,
                               const char *security_mech_id);
```

Check ID-WSF 2.0 Security Mechanism upon the received request. It is mandatory that a LassoServer is setted for the *profile* object.

Parameters

profile	a LassoIdWsf2Profile object	
security_mech_id	the security mechanism to enforce, if none is provided Bearer is assumed.	*[allow-none]*

Returns

0 if the request passed the check, an error code otherwise.

lasso_idwsf2_profile_get_name_identifier ()

```
LassoNode~*
lasso_idwsf2_profile_get_name_identifier
                               (LassoIdWsf2Profile *idwsf2_profile);
```

Return the NameIdentifier found in a WS-Security authentication token, when Bearer or SAML security mechanism is used. This method does not validate any security conditions on the assertion.

Parameters

idwsf2_profile	a LassoIdWsf2Profile object	

Returns

a LassoNode object or NULL.

[transfer full][allow-none]

lasso_idwsf2_profile_init_response ()

```
lasso_error_t
lasso_idwsf2_profile_init_response (LassoIdWsf2Profile *profile);
```

Initialize a new SOAP ID-WSF 2.0 response. Clear the existing response if one is currently set.

Parameters

profile	a LassoIdWsf2Profile object	

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_profile_init_soap_fault_response ()

```
lasso_error_t
lasso_idwsf2_profile_init_soap_fault_response
```

```
                               (LassoIdWsf2Profile *profile,
                                const char *faultcode,
                                const char *faultstring,
                                GList *details);
```

Initialize a new SOAP 1.1 fault.

Parameters

profile	a LassoIdWsf2Profile object	
faultcode	a SOAP fault code, see LASSO_SOAP_FAULT_CLIENT, LASSO_SOAP_FAULT_SERVER.	
faultstring	a human description of the error.	*[allow-none]*
details	complementary data describing the error, you can use LassoIdWsf2UtilStatus.	*[allow-none][element-type LassoNode]*

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_profile_redirect_user_for_interaction ()

```
lasso_error_t
lasso_idwsf2_profile_redirect_user_for_interaction
                               (LassoIdWsf2Profile *profile,
                                const gchar *redirect_url,
                                gboolean for_data);
```

Create a SOAP fault containing a RedirectRequest element, with a redirectURL property set to `redirect_url` concatenated with the parameter "transactionID" set to the messageID of the response message.

Parameters

profile	a LassoIdWsf2Profile object	
redirect_url	an URL where the user must be redirected	

Returns

0 if successful, an error code otherwise.

lasso_idwsf2_profile_build_response_msg ()

```
lasso_error_t
lasso_idwsf2_profile_build_response_msg
                               (LassoIdWsf2Profile *profile);
```

Serialize and sign the SOAP, if needed, the response message, put the result in

```
LASSO_PROFILE(profile)->msg_body
```

.

Parameters

idwsf2_profile	a LassoIdWsf2Profile object

Returns

0 if successful, LASSO_PROFILE_ERROR_BUILDING_RESPONSE_FAILED otherwise.

lasso_idwsf2_profile_process_response_msg ()

```
lasso_error_t
lasso_idwsf2_profile_process_response_msg
                               (LassoIdWsf2Profile *profile,
                                const char *msg);
```

Parse a response received by SOAP. Place the parsed message in the LassoIdWsf2Profile structure in the `soap_envelope_response` field and the content of the body in the `response` field.

Parameters

profile	a LassoIdWsf2Profile object
message	a string containing a response message

Returns

0 if successful, one of those error codes if the call fails:

- LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if first parameter is not a LassoIdWsf2Profile object,
- LASSO_PARAM_ERROR_INVALID_VALUE if message is NULL,
- LASSO_SOAP_ERROR_MISSING_BODY if no body element is found,
- LASSO_PROFILE_ERROR_MISSING_RESPONSE if the body element is empty.

Types and Values

struct LassoIdWsf2Profile

```
struct LassoIdWsf2Profile {
  LassoProfile parent;
};
```

10.5 Addon to SAML 2.0 profile for ID-WSF 2.0 bootsraping

Addon to SAML 2.0 profile for ID-WSF 2.0 bootsraping —

Functions

lasso_error_t	lasso_login_idwsf2_add_discovery_bootstrap_epr ()
LassoWsAddrEndpointReference *	lasso_login_idwsf2_get_discovery_bootstrap_epr ()
LassoWsAddrEndpointReference *	lasso_saml2_assertion_idwsf2_get_discovery_bootstrap_epr ()

Description

Functions

lasso_login_idwsf2_add_discovery_bootstrap_epr ()

```
lasso_error_t
lasso_login_idwsf2_add_discovery_bootstrap_epr
                                (LassoLogin *login,
                                 const char *url,
                                 const char *abstract,
                                 GList *security_mechanisms,
                                 int tolerance,
                                 int duration);
```

Add the needed bootstrap attribute to the LassoSaml2Assertion currently container in the LassoLogin object. This function should be called after lasso_login_build_assertion() by an IdP also having the Discovery service role.

The default `tolerance` and `duration` are respectively ten minutes and two days.

Parameters

login	a LassoLogin object	
url	the Disco service address	
abstract	the Disco service description	
security_mechanisms	the list of supported security mechanisms.	*[allow-none][element-type utf8]*
tolerance	see lasso_saml2_assertion_set_basic_conditions().	*[default -1]*
duration	see lasso_saml2_assertion_set_basic_conditions().	*[default 0]*

Returns

0 if successfull, otherwise LASSO_PROFILE_ERROR_MISSING_ASSERTION if no assertion is present in the LassoLogin object, LASSO_PARAM_ERROR_BAD_TYPE_OR_NULL_OBJ if login is not a LassoLogin object.

lasso_login_idwsf2_get_discovery_bootstrap_epr ()

```
LassoWsAddrEndpointReference~*
lasso_login_idwsf2_get_discovery_bootstrap_epr
                                (LassoLogin *login);
```

Extract the Discovery boostrap EPR from the attribute named LASSO_SAML2_ATTRIBUTE_NAME_EPR.

Parameters

login	a LassoLogin object

Returns

a caller owned LassoWsAddrEndpointReference object, or NULL if none can be found.

[transfer none]

lasso_saml2_assertion_idwsf2_get_discovery_bootstrap_epr ()

```
LassoWsAddrEndpointReference~*
lasso_saml2_assertion_idwsf2_get_discovery_bootstrap_epr
                               (LassoSaml2Assertion *assertion);
```

Extract the Discovery bootstrap EPR from `assertion` .

Parameters

assertion	a LassoSaml2Assertion object

Returns

a LassoWsAddrEndpointReference or NULL if no bootstrap EPR is found.

[transfer none]

Chapter 11

Objects from ID-WSF 2.0 schemas

11.1 Strings for ID-WSF 2.0

Strings for ID-WSF 2.0 —

Types and Values

#define	LASSO_IDWSF2_SB2_HREF
#define	LASSO_IDWSF2_SB2_PREFIX
#define	LASSO_IDWSF2_SBF_HREF
#define	LASSO_IDWSF2_SBF_PREFIX
#define	LASSO_IDWSF2_DISCOVERY_HREF
#define	LASSO_IDWSF2_DISCOVERY_PREFIX
#define	LASSO_IDWSF2_DST_HREF
#define	LASSO_IDWSF2_DST_PREFIX
#define	LASSO_IDWSF2_DSTREF_HREF
#define	LASSO_IDWSF2_DSTREF_PREFIX
#define	LASSO_IDWSF2_IMS_HREF
#define	LASSO_IDWSF2_IMS_PREFIX
#define	LASSO_IDWSF2_IS_HREF
#define	LASSO_IDWSF2_IS_PREFIX
#define	LASSO_IDWSF2_PS_HREF
#define	LASSO_IDWSF2_PS_PREFIX
#define	LASSO_IDWSF2_SUBS_HREF
#define	LASSO_IDWSF2_SUBS_PREFIX
#define	LASSO_IDWSF2_SUBSREF_HREF
#define	LASSO_IDWSF2_SUBSREF_PREFIX
#define	LASSO_IDWSF2_UTIL_HREF
#define	LASSO_IDWSF2_UTIL_PREFIX
#define	LASSO_IDWSF2_SEC_HREF
#define	LASSO_IDWSF2_SEC_PREFIX

Description

Functions

Types and Values

LASSO_IDWSF2_SB2_HREF

```
#define LASSO_IDWSF2_SB2_HREF "urn:liberty:sb:2006-08"
```

Namespace for ID-WSF 2.0 soap ninding

LASSO_IDWSF2_SB2_PREFIX

```
#define LASSO_IDWSF2_SB2_PREFIX "sb"
```

Preferred prefix for namespace of ID-WSF 2.0 soap binding

LASSO_IDWSF2_SBF_HREF

```
#define LASSO_IDWSF2_SBF_HREF "urn:liberty:sb"
```

Namespace for FIXME

LASSO_IDWSF2_SBF_PREFIX

```
#define LASSO_IDWSF2_SBF_PREFIX "sbf"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_DISCOVERY_HREF

```
#define LASSO_IDWSF2_DISCOVERY_HREF    "urn:liberty:disco:2006-08"
```

Namespace for ID-WSF 2.0 Discovery service

LASSO_IDWSF2_DISCOVERY_PREFIX

```
#define LASSO_IDWSF2_DISCOVERY_PREFIX "disco"
```

Preferred prefix for ID-WSF 2.0 Discovery service

LASSO_IDWSF2_DST_HREF

```
#define LASSO_IDWSF2_DST_HREF "urn:liberty:dst:2006-08"
```

Namespace for FIXME

LASSO_IDWSF2_DST_PREFIX

```
#define LASSO_IDWSF2_DST_PREFIX "dst"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_DSTREF_HREF

```
#define LASSO_IDWSF2_DSTREF_HREF "urn:liberty:dst:2006-08:ref"
```

Namespace for FIXME

LASSO_IDWSF2_DSTREF_PREFIX

```
#define LASSO_IDWSF2_DSTREF_PREFIX "dstref"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_IMS_HREF

```
#define LASSO_IDWSF2_IMS_HREF "urn:liberty:ims:2006-08"
```

Namespace for FIXME

LASSO_IDWSF2_IMS_PREFIX

```
#define LASSO_IDWSF2_IMS_PREFIX "ims"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_IS_HREF

```
#define LASSO_IDWSF2_IS_HREF "urn:liberty:is:2006-08"
```

Namespace for ID-WSF 2.0 Interaction Service

LASSO_IDWSF2_IS_PREFIX

```
#define LASSO_IDWSF2_IS_PREFIX "is"
```

Preferred prefix for namespace of ID-WSF 2.0 Interaction Service

LASSO_IDWSF2_PS_HREF

```
#define LASSO_IDWSF2_PS_HREF "urn:liberty:ps:2006-08"
```

Namespace for FIXME

LASSO_IDWSF2_PS_PREFIX

```
#define LASSO_IDWSF2_PS_PREFIX "ps"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_SUBS_HREF

```
#define LASSO_IDWSF2_SUBS_HREF "urn:liberty:ssos:2006-08"
```

Namespace for FIXME

LASSO_IDWSF2_SUBS_PREFIX

```
#define LASSO_IDWSF2_SUBS_PREFIX "subs"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_SUBSREF_HREF

```
#define LASSO_IDWSF2_SUBSREF_HREF "urn:liberty:ssos:2006-08:ref"
```

Namespace for ID-WSF 2.0 subscription service

LASSO_IDWSF2_SUBSREF_PREFIX

```
#define LASSO_IDWSF2_SUBSREF_PREFIX "subsref"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_UTIL_HREF

```
#define LASSO_IDWSF2_UTIL_HREF "urn:liberty:util:2006-08"
```

Namespace for FIXME

LASSO_IDWSF2_UTIL_PREFIX

```
#define LASSO_IDWSF2_UTIL_PREFIX "util"
```

Preferred prefix for namespace of FIXME

LASSO_IDWSF2_SEC_HREF

```
#define LASSO_IDWSF2_SEC_HREF "urn:liberty:security:2006-08"
```

Namespace for FIXME

LASSO_IDWSF2_SEC_PREFIX

```
#define LASSO_IDWSF2_SEC_PREFIX "sec"
```

Preferred prefix for namespace of FIXME

11.2 LassoIdWsf2DiscoServiceType

LassoIdWsf2DiscoServiceType — <disco:Service>

Functions

LassoIdWsf2DiscoServiceType *	lasso_idwsf2_disco_service_type_new ()
LassoIdWsf2DiscoServiceType *	lasso_idwsf2_disco_service_type_new_with_string ()

Types and Values

struct	LassoIdWsf2DiscoServiceType

Description

```

<xs:element name="ServiceType" type="xs:anyURI"/>
```

Figure 11.1: Schema fragment for disco:Service

Functions

lasso_idwsf2_disco_service_type_new ()

```
LassoIdWsf2DiscoServiceType~*
lasso_idwsf2_disco_service_type_new (void);
```

Creates a new LassoIdWsf2DiscoServiceType object.

Returns

a newly created LassoIdWsf2DiscoServiceType object

lasso_idwsf2_disco_service_type_new_with_string ()

```
LassoIdWsf2DiscoServiceType~*
lasso_idwsf2_disco_service_type_new_with_string
                                        (const char *content);
```

Creates a new LassoIdWsf2DiscoServiceType object and initializes it with `content` as content.

Parameters

content	the content string

Returns

a newly created LassoIdWsf2DiscoServiceType object

Types and Values

struct LassoIdWsf2DiscoServiceType

```
struct LassoIdWsf2DiscoServiceType {
  LassoNode parent;

  /* elements */
  char *content;
};
```

11.3 LassoIdWsf2DiscoAbstract

LassoIdWsf2DiscoAbstract — <disco:Abstract>

Functions

LassoIdWsf2DiscoAbstract *	lasso_idwsf2_disco_abstract_new ()
LassoIdWsf2DiscoAbstract *	lasso_idwsf2_disco_abstract_new_with_string ()

Types and Values

struct	LassoIdWsf2DiscoAbstract

Description

```
<xs:element name="Abstract" type="xs:string"/>
```

Figure 11.2: Schema fragment for disco:Abstract

Functions

lasso_idwsf2_disco_abstract_new ()

```
LassoIdWsf2DiscoAbstract~*
lasso_idwsf2_disco_abstract_new (void);
```

Creates a new LassoIdWsf2DiscoAbstract object.

Returns

a newly created LassoIdWsf2DiscoAbstract object

lasso_idwsf2_disco_abstract_new_with_string ()

```
LassoIdWsf2DiscoAbstract~*
lasso_idwsf2_disco_abstract_new_with_string
                               (const char *content);
```

Creates a new LassoIdWsf2DiscoAbstract object and initializes it with `content` as content.

Parameters

content	the content string

Returns

a newly created LassoIdWsf2DiscoAbstract object

Types and Values

struct LassoIdWsf2DiscoAbstract

```
struct LassoIdWsf2DiscoAbstract {
  LassoNode parent;

  /* elements */
  char *content;
};
```

11.4 LassoIdWsf2DiscoEndpointContext

LassoIdWsf2DiscoEndpointContext — <disco:EndpointContext>

Functions

LassoIdWsf2DiscoEndpointContext *	lasso_idwsf2_disco_endpoint_context_new ()
LassoIdWsf2DiscoEndpointContext *	lasso_idwsf2_disco_endpoint_context_new_full ()

Types and Values

struct	LassoIdWsf2DiscoEndpointContext

Description

```
<xs:complexType name="EndpointContextType">
  <xs:sequence>
    <xs:element ref="Address"        maxOccurs="unbounded" />
    <xs:element ref="sbf:Framework"  maxOccurs="unbounded" />
    <xs:element ref="SecurityMechID" maxOccurs="unbounded" />
    <xs:element ref="Action"         minOccurs="0"
      maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
```

Figure 11.3: Schema fragment for disco:EndpointContext

Functions

lasso_idwsf2_disco_endpoint_context_new ()

```
LassoIdWsf2DiscoEndpointContext~*
lasso_idwsf2_disco_endpoint_context_new
                               (void);
```

Creates a new LassoIdWsf2DiscoEndpointContext object.

Returns

a newly created LassoIdWsf2DiscoEndpointContext object

lasso_idwsf2_disco_endpoint_context_new_full ()

```
LassoIdWsf2DiscoEndpointContext~*
lasso_idwsf2_disco_endpoint_context_new_full
                               (const gchar *address);
```

Types and Values

struct LassoIdWsf2DiscoEndpointContext

```
struct LassoIdWsf2DiscoEndpointContext {
  LassoNode parent;

  /* elements */
  GList *Address; /* of strings */
  GList *Framework; /* of LassoNode */
  GList *SecurityMechID; /* of strings */
  GList *Action; /* of strings */
};
```

11.5 LassoIdWsf2DiscoKeys

LassoIdWsf2DiscoKeys — <disco:Keys>

Functions

LassoIdWsf2DiscoKeys *	lasso_idwsf2_disco_keys_new ()

Types and Values

struct	LassoIdWsf2DiscoKeys

Description

```
<xs:complexType name="KeysType">
  <xs:sequence>
    <xs:element ref="md:KeyDescriptor"
      minOccurs="1"
      maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

Figure 11.4: Schema fragment for disco:Keys

Functions

lasso_idwsf2_disco_keys_new ()

```
LassoIdWsf2DiscoKeys~*
lasso_idwsf2_disco_keys_new (void);
```

Creates a new LassoIdWsf2DiscoKeys object.

Returns

a newly created LassoIdWsf2DiscoKeys object

Types and Values

struct LassoIdWsf2DiscoKeys

```
struct LassoIdWsf2DiscoKeys {
  LassoNode parent;

  /* elements */
  GList *KeyDescriptor; /* of LassoNode */
};
```

11.6 LassoIdWsf2DiscoProviderID

LassoIdWsf2DiscoProviderID — <disco:ProviderID>

Functions

LassoIdWsf2DiscoProviderID *	lasso_idwsf2_disco_provider_id_new ()
LassoIdWsf2DiscoProviderID *	lasso_idwsf2_disco_provider_id_new_with_string ()

Types and Values

struct	LassoIdWsf2DiscoProviderID

Description

```
<xs:element name="ProviderID" type="xs:anyURI"/>
```

Figure 11.5: Schema fragment for disco:ProviderID

Functions

lasso_idwsf2_disco_provider_id_new ()

```
LassoIdWsf2DiscoProviderID~*
lasso_idwsf2_disco_provider_id_new (void);
```

Creates a new LassoIdWsf2DiscoProviderID object.

Returns

a newly created LassoIdWsf2DiscoProviderID object

lasso_idwsf2_disco_provider_id_new_with_string ()

```
LassoIdWsf2DiscoProviderID~*
lasso_idwsf2_disco_provider_id_new_with_string
                                (const char *content);
```

Creates a new LassoIdWsf2DiscoProviderID object and initializes it with `content` as content.

Parameters

content	the content string

Returns

a newly created LassoIdWsf2DiscoProviderID object

Types and Values

struct LassoIdWsf2DiscoProviderID

```
struct LassoIdWsf2DiscoProviderID {
  LassoNode parent;

  /* elements */
  char *content;
};
```

11.7 LassoIdWsf2DiscoRequestedService

LassoIdWsf2DiscoRequestedService — <disco:RequestedService>

Functions

LassoIdWsf2DiscoRequestedService *	lasso_idwsf2_disco_requested_service_new ()

Types and Values

struct	LassoIdWsf2DiscoRequestedService

Description

```
<xs:complexType name="RequestedServiceType">
  <xs:sequence>
    <xs:element ref="ServiceType" minOccurs="0" maxOccurs="unbounded" />

    <xs:element ref="ProviderID" minOccurs="0" maxOccurs="unbounded" />

    <xs:element ref="Options" minOccurs="0" maxOccurs="unbounded"/>

    <xs:element ref="SecurityMechID" minOccurs="0" maxOccurs="unbounded"/>

    <xs:element ref="Framework" minOccurs="0" maxOccurs="unbounded"/>

    <xs:element ref="Action" minOccurs="0" maxOccurs="unbounded"/>

    <xs:any namespace="##other"
      processContents="lax"
      minOccurs="0"
      maxOccurs="unbounded"/>

    </xs:sequence>

    <xs:attribute name="reqID" type="xs:string" use="optional" />
    <xs:attribute name="resultsType" type="xs:string" use="optional" />

  </xs:complexType>
```

Figure 11.6: Schema fragment for disco:RequestedService

Functions

lasso_idwsf2_disco_requested_service_new ()

```
LassoIdWsf2DiscoRequestedService~*
lasso_idwsf2_disco_requested_service_new
                               (void);
```

Creates a new LassoIdWsf2DiscoRequestedService object.

Returns

a newly created LassoIdWsf2DiscoRequestedService object

Types and Values

struct LassoIdWsf2DiscoRequestedService

```
struct LassoIdWsf2DiscoRequestedService {
  LassoNode parent;

  /* elements */
  GList *ServiceType; /* of strings */
  GList *ProviderID; /* of strings */
  GList *Options; /* of LassoIdWsf2DiscoOptions */
  GList *SecurityMechID; /* of strings */
  GList *Framework; /* of LassoIdWsf2SbfFramework */
  GList *Action; /* of strings */
  LassoNode *any;
  /* attributes */
  char *reqID;
  char *resultsType;
};
```

11.8 LassoIdWsf2DiscoSecurityContext

LassoIdWsf2DiscoSecurityContext — <disco:SecurityContext>

Functions

LassoIdWsf2DiscoSecurityContext *	lasso_idwsf2_disco_security_context_new ()

Types and Values

struct	LassoIdWsf2DiscoSecurityContext

Description

```
<xs:element name="SecurityContext">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="SecurityMechID"
        minOccurs="1"
        maxOccurs="unbounded"/>

        <xs:element ref="sec:Token"
          minOccurs="0"
          maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
```

Figure 11.7: Schema fragment for disco:SecurityContext

Functions

lasso_idwsf2_disco_security_context_new ()

```
LassoIdWsf2DiscoSecurityContext~*
lasso_idwsf2_disco_security_context_new
                                (void);
```

Creates a new LassoIdWsf2DiscoSecurityContext object.

Returns

a newly created LassoIdWsf2DiscoSecurityContext object

Types and Values

struct LassoIdWsf2DiscoSecurityContext

```
struct LassoIdWsf2DiscoSecurityContext {
  LassoNode parent;

  /* elements */
  GList *SecurityMechID; /* of strings */
  GList *Token; /* of LassoNode */
};
```

11.9 LassoIdWsf2DiscoServiceContext

LassoIdWsf2DiscoServiceContext — <disco:ServiceContext>

Functions

LassoIdWsf2DiscoServiceContext *	lasso_idwsf2_disco_service_context_new ()
LassoIdWsf2DiscoServiceContext *	lasso_idwsf2_disco_service_context_new_full ()

Types and Values

struct	LassoIdWsf2DiscoServiceContext

Description

```
<xs:complexType name="ServiceContextType">
  <xs:sequence>
    <xs:element ref="ServiceType"     maxOccurs="unbounded" />
    <xs:element ref="Options"         minOccurs="0"
      maxOccurs="unbounded" />
      <xs:element ref="EndpointContext" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
```

Figure 11.8: Schema fragment for disco:ServiceContext

Functions

lasso_idwsf2_disco_service_context_new ()

```
LassoIdWsf2DiscoServiceContext~*
```

```
lasso_idwsf2_disco_service_context_new
                                (void);
```

Creates a new LassoIdWsf2DiscoServiceContext object.

Returns

a newly created LassoIdWsf2DiscoServiceContext object

lasso_idwsf2_disco_service_context_new_full ()

```
LassoIdWsf2DiscoServiceContext~*
lasso_idwsf2_disco_service_context_new_full
                                (const gchar *serviceType,
                                 LassoIdWsf2DiscoEndpointContext *endpointContext);
```

Types and Values

struct LassoIdWsf2DiscoServiceContext

```
struct LassoIdWsf2DiscoServiceContext {
  LassoNode parent;

  /* elements */
  GList *ServiceType; /* of strings */
  GList *Options; /* of LassoIdWsf2DiscoOptions */
  GList *EndpointContext; /* of LassoIdWsf2DiscoEndpointContext */
};
```

11.10 LassoIdWsf2DiscoSvcMDAssociationAddResponse

LassoIdWsf2DiscoSvcMDAssociationAddResponse — <disco:SvcMDAssociationAddResponse>

Functions

LassoIdWsf2DiscoSvcMDAssociationAddResponse *	lasso_idwsf2_disco_svc_md_association_add_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDAssociationAddResponse

Description

```
<xs:complexType name="SvcMDAssociationAddResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.9: Schema fragment for disco:SvcMDAssociationAddResponse

Functions

lasso_idwsf2_disco_svc_md_association_add_response_new ()

```
LassoIdWsf2DiscoSvcMDAssociationAddResponse~*
lasso_idwsf2_disco_svc_md_association_add_response_new
                               (void);
```

Creates a new LassoIdWsf2DiscoSvcMDAssociationAddResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDAssociationAddResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDAssociationAddResponse

```
struct LassoIdWsf2DiscoSvcMDAssociationAddResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  /* attributes */
  GHashTable *attributes;
};
```

11.11 LassoIdWsf2DiscoSvcMDAssociationAdd

LassoIdWsf2DiscoSvcMDAssociationAdd — <disco:SvcMDAssociationAdd>

Functions

LassoIdWsf2DiscoSvcMDAssociationAdd *	lasso_idwsf2_disco_svc_md_association_add_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDAssociationAdd

Description

```
<xs:complexType name="SvcMDAssociationAddType">
  <xs:sequence>
    <xs:element ref="SvcMDID" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.10: Schema fragment for disco:SvcMDAssociationAdd

Functions

lasso_idwsf2_disco_svc_md_association_add_new ()

```
LassoIdWsf2DiscoSvcMDAssociationAdd~*
lasso_idwsf2_disco_svc_md_association_add_new
                                (void);
```

Creates a new LassoIdWsf2DiscoSvcMDAssociationAdd object.

Returns

a newly created LassoIdWsf2DiscoSvcMDAssociationAdd object

Types and Values

struct LassoIdWsf2DiscoSvcMDAssociationAdd

```
struct LassoIdWsf2DiscoSvcMDAssociationAdd {
  LassoNode parent;

  /* elements */
  GList *SvcMDID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.12 LassoIdWsf2DiscoSvcMDAssociationDeleteResponse

LassoIdWsf2DiscoSvcMDAssociationDeleteResponse — <disco:SvcMDAssociationDeleteResponse>

Functions

LassoIdWsf2DiscoSvcMDAssociationDeleteResponse *	lasso_idwsf2_disco_svc_md_association_delete_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDAssociationDeleteResponse

Description

```
<xs:complexType name="SvcMDAssociationDeleteResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.11: Schema fragment for disco:SvcMDAssociationDeleteResponse

Functions

lasso_idwsf2_disco_svc_md_association_delete_response_new ()

```
LassoIdWsf2DiscoSvcMDAssociationDeleteResponse~*
lasso_idwsf2_disco_svc_md_association_delete_response_new
                                (void);
```

Creates a new LassoIdWsf2DiscoSvcMDAssociationDeleteResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDAssociationDeleteResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDAssociationDeleteResponse

```
struct LassoIdWsf2DiscoSvcMDAssociationDeleteResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  /* attributes */
  GHashTable *attributes;
};
```

11.13 LassoIdWsf2DiscoSvcMDAssociationDelete

LassoIdWsf2DiscoSvcMDAssociationDelete — <disco:SvcMDAssociationDelete>

Functions

LassoIdWsf2DiscoSvcMDAssociationDelete *	lasso_idwsf2_disco_svc_md_association_delete_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDAssociationDelete

Description

```
<xs:complexType name="SvcMDAssociationDeleteType">
  <xs:sequence>
    <xs:element ref="SvcMDID" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.12: Schema fragment for disco:SvcMDAssociationDelete

Functions

lasso_idwsf2_disco_svc_md_association_delete_new ()

```
LassoIdWsf2DiscoSvcMDAssociationDelete~*
lasso_idwsf2_disco_svc_md_association_delete_new
                                (void);
```

Creates a new LassoIdWsf2DiscoSvcMDAssociationDelete object.

Returns

a newly created LassoIdWsf2DiscoSvcMDAssociationDelete object

Types and Values

struct LassoIdWsf2DiscoSvcMDAssociationDelete

```
struct LassoIdWsf2DiscoSvcMDAssociationDelete {
  LassoNode parent;

  /* elements */
  GList *SvcMDID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.14 LassoIdWsf2DiscoSvcMDAssociationQueryResponse

LassoIdWsf2DiscoSvcMDAssociationQueryResponse — <disco:SvcMDAssociationQueryResponse>

Functions

LassoIdWsf2DiscoSvcMDAssociationQueryResponse *	lasso_idwsf2_disco_svc_md_association_query_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDAssociationQueryResponse

Description

```
<xs:complexType name="SvcMDAssociationQueryResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status" />
    <xs:element ref="SvcMDID" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.13: Schema fragment for disco:SvcMDAssociationQueryResponse

Functions

lasso_idwsf2_disco_svc_md_association_query_response_new ()

```
LassoIdWsf2DiscoSvcMDAssociationQueryResponse~*
lasso_idwsf2_disco_svc_md_association_query_response_new
                               (void);
```

Creates a new LassoIdWsf2DiscoSvcMDAssociationQueryResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDAssociationQueryResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDAssociationQueryResponse

```
struct LassoIdWsf2DiscoSvcMDAssociationQueryResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  GList *SvcMDID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.15 LassoIdWsf2DiscoSvcMDAssociationQuery

LassoIdWsf2DiscoSvcMDAssociationQuery — <disco:SvcMDAssociationQuery>

Functions

LassoIdWsf2DiscoSvcMDAssociationQuery *	lasso_idwsf2_disco_svc_md_association_query_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDAssociationQuery

Description

```
<xs:complexType name="SvcMDAssociationQueryType">
  <xs:sequence>
    <xs:element ref="SvcMDID" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.14: Schema fragment for disco:SvcMDAssociationQuery

Functions

lasso_idwsf2_disco_svc_md_association_query_new ()

```
LassoIdWsf2DiscoSvcMDAssociationQuery~*
lasso_idwsf2_disco_svc_md_association_query_new
                               (void);
```

Creates a new LassoIdWsf2DiscoSvcMDAssociationQuery object.

Returns

a newly created LassoIdWsf2DiscoSvcMDAssociationQuery object

Types and Values

struct LassoIdWsf2DiscoSvcMDAssociationQuery

```
struct LassoIdWsf2DiscoSvcMDAssociationQuery {
  LassoNode parent;

  /* elements */
  GList *SvcMDID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.16 LassoIdWsf2DiscoSvcMDDeleteResponse

LassoIdWsf2DiscoSvcMDDeleteResponse — <disco:SvcMDDeleteResponse>

Functions

LassoIdWsf2DiscoSvcMDDeleteResponse *	lasso_idwsf2_disco_svc_md_delete_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDDeleteResponse

Description

```
<xs:complexType name="SvcMDDeleteResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.15: Schema fragment for disco:SvcMDDeleteResponse

Functions

lasso_idwsf2_disco_svc_md_delete_response_new ()

```
LassoIdWsf2DiscoSvcMDDeleteResponse~*
lasso_idwsf2_disco_svc_md_delete_response_new
                               (void);
```

Creates a new LassoIdWsf2DiscoSvcMDDeleteResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDDeleteResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDDeleteResponse

```
struct LassoIdWsf2DiscoSvcMDDeleteResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  /* attributes */
  GHashTable *attributes;
};
```

11.17 LassoIdWsf2DiscoSvcMDDelete

LassoIdWsf2DiscoSvcMDDelete — <disco:SvcMDDelete>

Functions

LassoIdWsf2DiscoSvcMDDelete *	lasso_idwsf2_disco_svc_md_delete_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDDelete

Description

```
<xs:complexType name="SvcMDDeleteType">
  <xs:sequence>
    <xs:element ref="SvcMDID" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.16: Schema fragment for disco:SvcMDDelete

Functions

lasso_idwsf2_disco_svc_md_delete_new ()

```
LassoIdWsf2DiscoSvcMDDelete~*
lasso_idwsf2_disco_svc_md_delete_new (void);
```

Creates a new LassoIdWsf2DiscoSvcMDDelete object.

Returns

a newly created LassoIdWsf2DiscoSvcMDDelete object

Types and Values

struct LassoIdWsf2DiscoSvcMDDelete

```
struct LassoIdWsf2DiscoSvcMDDelete {
  LassoNode parent;

  /* elements */
  GList *SvcMDID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.18 LassoIdWsf2DiscoSvcMDQueryResponse

LassoIdWsf2DiscoSvcMDQueryResponse — <disco:SvcMDQueryResponse>

Functions

LassoIdWsf2DiscoSvcMDQueryResponse *	lasso_idwsf2_disco_svc_md_query_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDQueryResponse

Description

```
<xs:complexType name="SvcMDQueryResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status" />
    <xs:element ref="SvcMD" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.17: Schema fragment for disco:SvcMDQueryResponse

Functions

lasso_idwsf2_disco_svc_md_query_response_new ()

```
LassoIdWsf2DiscoSvcMDQueryResponse~*
lasso_idwsf2_disco_svc_md_query_response_new
                                (void);
```

Creates a new LassoIdWsf2DiscoSvcMDQueryResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDQueryResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDQueryResponse

```
struct LassoIdWsf2DiscoSvcMDQueryResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  GList *SvcMD; /* of LassoIdWsf2DiscoSvcMetadata */
  /* attributes */
  GHashTable *attributes;
};
```

11.19 LassoIdWsf2DiscoSvcMDQuery

LassoIdWsf2DiscoSvcMDQuery — <disco:SvcMDQuery>

Functions

LassoIdWsf2DiscoSvcMDQuery *	lasso_idwsf2_disco_svc_md_query_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDQuery

Description

```
<xs:complexType name="SvcMDQueryType">
  <xs:sequence>
    <xs:element ref="SvcMDID"
      minOccurs="0"
      maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

Figure 11.18: Schema fragment for disco:SvcMDQuery

Functions

lasso_idwsf2_disco_svc_md_query_new ()

```
LassoIdWsf2DiscoSvcMDQuery~*
lasso_idwsf2_disco_svc_md_query_new (void);
```

Creates a new LassoIdWsf2DiscoSvcMDQuery object.

Returns

a newly created LassoIdWsf2DiscoSvcMDQuery object

Types and Values

struct LassoIdWsf2DiscoSvcMDQuery

```
struct LassoIdWsf2DiscoSvcMDQuery {
  LassoNode parent;

  /* elements */
  GList *SvcMDID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.20 LassoIdWsf2DiscoSvcMDRegisterResponse

LassoIdWsf2DiscoSvcMDRegisterResponse — <disco:SvcMDRegisterResponse>

Functions

LassoIdWsf2DiscoSvcMDRegisterResponse *	lasso_idwsf2_disco_svc_md_register_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDRegisterResponse

Description

```
<xs:complexType name="SvcMDRegisterResponseType">
  <xs:sequence>

    <xs:element ref="lu:Status" />
    <xs:element ref="SvcMDID"   minOccurs="0" maxOccurs="unbounded" />
    <xs:element ref="Keys"      minOccurs="0" maxOccurs="unbounded" />

  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.19: Schema fragment for disco:SvcMDRegisterResponse

Functions

lasso_idwsf2_disco_svc_md_register_response_new ()

```
LassoIdWsf2DiscoSvcMDRegisterResponse~*
lasso_idwsf2_disco_svc_md_register_response_new
                                (void);
```

Creates a new LassoIdWsf2DiscoSvcMDRegisterResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDRegisterResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDRegisterResponse

```
struct LassoIdWsf2DiscoSvcMDRegisterResponse {
  LassoNode parent;
  /* elements */
  LassoIdWsf2UtilStatus *Status;
  GList *SvcMDID; /* of strings */
  GList *Keys; /* of LassoIdWsf2DiscoKeys */
  /* attributes */
  GHashTable *attributes;
};
```

11.21 LassoIdWsf2DiscoSvcMDRegister

LassoIdWsf2DiscoSvcMDRegister — <disco:SvcMDRegister>

Functions

LassoIdWsf2DiscoSvcMDRegister *	lasso_idwsf2_disco_svc_md_register_new ()
LassoIdWsf2DiscoSvcMDRegister *	lasso_idwsf2_disco_svc_md_register_new_full ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDRegister

Description

```
<xs:complexType name="SvcMDRegisterType">
  <xs:sequence>
    <xs:element ref="SvcMD" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.20: Schema fragment for disco:SvcMDRegister

Functions

lasso_idwsf2_disco_svc_md_register_new ()

```
LassoIdWsf2DiscoSvcMDRegister~*
lasso_idwsf2_disco_svc_md_register_new
                               (void);
```

Creates a new LassoIdWsf2DiscoSvcMDRegister object.

Returns

a newly created LassoIdWsf2DiscoSvcMDRegister object

lasso_idwsf2_disco_svc_md_register_new_full ()

```
LassoIdWsf2DiscoSvcMDRegister~*
lasso_idwsf2_disco_svc_md_register_new_full
                               (const gchar *service_type,
                                const gchar *abstract,
                                const gchar *provider_id,
                                const gchar *soap_endpoint);
```

Create and initialize a complete message for registering new metadatas at a discovery service.

Parameters

service_type	the service type for the registered metadatas	
abstract	the human description for the service	
provider_id	the SAML provider id of the service	
soap_endpoint	the SOAP endpoint URL for the service	

Returns

a new filled and initialized LassoIdWsf2DiscoSvcMDRegister if successfull, NULL otherwise.

Types and Values

struct LassoIdWsf2DiscoSvcMDRegister

```
struct LassoIdWsf2DiscoSvcMDRegister {
  LassoNode parent;

  /* elements */
  GList *SvcMD; /* of LassoIdWsf2DiscoSvcMetadata */
  /* attributes */
  GHashTable *attributes;
};
```

11.22 LassoIdWsf2DiscoSvcMDReplaceResponse

LassoIdWsf2DiscoSvcMDReplaceResponse — <disco:SvcMDReplaceResponse>

Functions

LassoIdWsf2DiscoSvcMDReplaceResponse *	lasso_idwsf2_disco_svc_md_replace_response_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDReplaceResponse

Description

```
<xs:complexType name="SvcMDReplaceResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.21: Schema fragment for disco:SvcMDReplaceResponse

Functions

lasso_idwsf2_disco_svc_md_replace_response_new ()

```
LassoIdWsf2DiscoSvcMDReplaceResponse~*
lasso_idwsf2_disco_svc_md_replace_response_new
                               (void);
```

Creates a new LassoIdWsf2DiscoSvcMDReplaceResponse object.

Returns

a newly created LassoIdWsf2DiscoSvcMDReplaceResponse object

Types and Values

struct LassoIdWsf2DiscoSvcMDReplaceResponse

```
struct LassoIdWsf2DiscoSvcMDReplaceResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  /* attributes */
  GHashTable *attributes;
};
```

11.23 LassoIdWsf2DiscoSvcMDReplace

LassoIdWsf2DiscoSvcMDReplace — <disco:SvcMDReplace>

Functions

LassoIdWsf2DiscoSvcMDReplace *	lasso_idwsf2_disco_svc_md_replace_new ()

Types and Values

struct	LassoIdWsf2DiscoSvcMDReplace

Description

```
<xs:complexType name="SvcMDReplaceType">
  <xs:sequence>
    <xs:element ref="SvcMD" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.22: Schema fragment for disco:SvcMDReplace

Functions

lasso_idwsf2_disco_svc_md_replace_new ()

```
LassoIdWsf2DiscoSvcMDReplace~*
lasso_idwsf2_disco_svc_md_replace_new (void);
```

Creates a new LassoIdWsf2DiscoSvcMDReplace object.

Returns

a newly created LassoIdWsf2DiscoSvcMDReplace object

Types and Values

struct LassoIdWsf2DiscoSvcMDReplace

```
struct LassoIdWsf2DiscoSvcMDReplace {
  LassoNode parent;

  /* elements */
  GList *SvcMD; /* of LassoIdWsf2DiscoSvcMetadata */
  /* attributes */
  GHashTable *attributes;
};
```

11.24 LassoIdWsf2DiscoSvcMetadata

LassoIdWsf2DiscoSvcMetadata — <disco:SvcMetadata>

Functions

LassoIdWsf2DiscoSvcMetadata *	lasso_idwsf2_disco_svc_metadata_new ()
LassoIdWsf2DiscoSvcMetadata *	lasso_idwsf2_disco_svc_metadata_new_full ()

Types and Values

struct	LassoIdWsf2DiscoSvcMetadata

Description

```
<xs:complexType name="SvcMetadataType">
  <xs:sequence>
    <xs:element ref="Abstract"                              />
    <xs:element ref="ProviderID"                            />
    <xs:element ref="ServiceContext"  maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="svcMDID" type="xs:string" use="optional" />
</xs:complexType>
```

Figure 11.23: Schema fragment for disco:SvcMetadata

Functions

lasso_idwsf2_disco_svc_metadata_new ()

```
LassoIdWsf2DiscoSvcMetadata~*
lasso_idwsf2_disco_svc_metadata_new (void);
```

Creates a new LassoIdWsf2DiscoSvcMetadata object.

Returns

a newly created LassoIdWsf2DiscoSvcMetadata object

lasso_idwsf2_disco_svc_metadata_new_full ()

```
LassoIdWsf2DiscoSvcMetadata~*
lasso_idwsf2_disco_svc_metadata_new_full
                               (const gchar *service_type,
                                const gchar *abstract,
                                const gchar *provider_id,
                                const gchar *soap_endpoint);
```

Types and Values

struct LassoIdWsf2DiscoSvcMetadata

```
struct LassoIdWsf2DiscoSvcMetadata {
  LassoNode parent;

  /* elements */
  char *Abstract;
  char *ProviderID;
  GList *ServiceContext; /* of LassoIdWsf2DiscoServiceContext */
  /* attributes */
  char *svcMDID;
};
```

11.25 LassoIdWsf2DstDataResponseBase

LassoIdWsf2DstDataResponseBase — <dst:DataResponseBase>

Functions

LassoIdWsf2DstDataResponseBase *	lasso_idwsf2_dst_data_response_base_new ()

Types and Values

struct	LassoIdWsf2DstDataResponseBase

Description

```
<xs:complexType name="DataResponseBaseType">
  <xs:complexContent>
    <xs:extension base="lu:ResponseType">
      <xs:attribute name="timeStamp" use="optional" type="xs:dateTime"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.24: Schema fragment for dst:DataResponseBase

Functions

lasso_idwsf2_dst_data_response_base_new ()

```
LassoIdWsf2DstDataResponseBase~*
lasso_idwsf2_dst_data_response_base_new
                                (void);
```

Creates a new LassoIdWsf2DstDataResponseBase object.

Returns

a newly created LassoIdWsf2DstDataResponseBase object

Types and Values

struct LassoIdWsf2DstDataResponseBase

```
struct LassoIdWsf2DstDataResponseBase {
  LassoIdWsf2UtilResponse parent;

  /* attributes */
  char *timeStamp;
};
```

11.26 LassoIdWsf2DstDeleteItemBase

LassoIdWsf2DstDeleteItemBase — <dst:DeleteItemBase>

Functions

LassoIdWsf2DstDeleteItemBase *	lasso_idwsf2_dst_delete_item_base_new ()

Types and Values

struct	LassoIdWsf2DstDeleteItemBase

Description

```
<xs:complexType name="DeleteItemBaseType">
  <xs:attributeGroup ref="dst:selectQualif"/>
  <xs:attribute name="notChangedSince" use="optional" type="xs:dateTime"/>
  <xs:attribute name="id" use="optional" type="xs:ID"/>
  <xs:attribute ref="lu:itemID" use="optional"/>
</xs:complexType>
```

Figure 11.25: Schema fragment for dst:DeleteItemBase

Functions

lasso_idwsf2_dst_delete_item_base_new ()

```
LassoIdWsf2DstDeleteItemBase~*
lasso_idwsf2_dst_delete_item_base_new (void);
```

Creates a new LassoIdWsf2DstDeleteItemBase object.

Returns

a newly created LassoIdWsf2DstDeleteItemBase object

Types and Values

struct LassoIdWsf2DstDeleteItemBase

```
struct LassoIdWsf2DstDeleteItemBase {
  LassoNode parent;

  /* attributes */
  char *notChangedSince;
  char *id;
  char *itemID;
  char *objectType;
  char *predefined;
};
```

11.27 LassoIdWsf2DstDeleteResponse

LassoIdWsf2DstDeleteResponse — <dst:DeleteResponse>

Functions

LassoIdWsf2DstDeleteResponse *	lasso_idwsf2_dst_delete_response_new ()

Types and Values

struct	LassoIdWsf2DstDeleteResponse

Description

```

<xs:complexType name="DeleteResponseType">
  <xs:complexContent>
    <xs:extension base="lu:ResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.26: Schema fragment for dst:DeleteResponse

Functions

lasso_idwsf2_dst_delete_response_new ()

```
LassoIdWsf2DstDeleteResponse~*
lasso_idwsf2_dst_delete_response_new (void);
```

Creates a new LassoIdWsf2DstDeleteResponse object.

Returns

a newly created LassoIdWsf2DstDeleteResponse object

Types and Values

struct LassoIdWsf2DstDeleteResponse

```
struct LassoIdWsf2DstDeleteResponse {
  LassoIdWsf2UtilResponse parent;
};
```

11.28 LassoIdWsf2DstRefAppData

LassoIdWsf2DstRefAppData — <dstref:AppData>

Functions

LassoIdWsf2DstRefAppData *	lasso_idwsf2_dstref_app_data_new ()

Types and Values

struct	LassoIdWsf2DstRefAppData

Description

```

<xs:complexType name="AppDataType">
  <xs:simpleContent>
    <xs:extension base="xs:string"/>
  </xs:simpleContent>
</xs:complexType>
```

Figure 11.27: Schema fragment for dstref:AppData

Functions

lasso_idwsf2_dstref_app_data_new ()

```
LassoIdWsf2DstRefAppData~*
lasso_idwsf2_dstref_app_data_new (void);
```

Creates a new LassoIdWsf2DstRefAppData object.

Returns

a newly created LassoIdWsf2DstRefAppData object

Types and Values

struct LassoIdWsf2DstRefAppData

```
struct LassoIdWsf2DstRefAppData {
  LassoNode parent;

  /* elements */
  GList *any; /* of xmlNode* */
};
```

11.29 LassoIdWsf2DstRefCreateItem

LassoIdWsf2DstRefCreateItem — <dstref:CreateItem>

Functions

LassoIdWsf2DstRefCreateItem *	lasso_idwsf2_dstref_create_item_new ()

Types and Values

struct	LassoIdWsf2DstRefCreateItem

Description

```

<xs:complexType name="CreateItemType">
  <xs:sequence>
    <xs:element ref="dstref:NewData" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="dst:CreateItemAttributeGroup"/>
</xs:complexType>
```

Figure 11.28: Schema fragment for dstref:CreateItem

Functions

lasso_idwsf2_dstref_create_item_new ()

```
LassoIdWsf2DstRefCreateItem~*
lasso_idwsf2_dstref_create_item_new (void);
```

Creates a new LassoIdWsf2DstRefCreateItem object.

Returns

a newly created LassoIdWsf2DstRefCreateItem object

Types and Values

struct LassoIdWsf2DstRefCreateItem

```
struct LassoIdWsf2DstRefCreateItem {
  LassoNode parent;

  /* elements */
  LassoIdWsf2DstRefAppData *NewData;
  /* attributes */
  char *objectType;
  char *id;
  char *itemID;
};
```

11.30 LassoIdWsf2DstRefCreateResponse

LassoIdWsf2DstRefCreateResponse — <dstref:CreateResponse>

Functions

LassoIdWsf2DstRefCreateResponse *	lasso_idwsf2_dstref_create_response_new ()

Types and Values

struct	LassoIdWsf2DstRefCreateResponse

Description

```

<xs:complexType name="CreateResponseType">
  <xs:complexContent>
    <xs:extension base="dstref:DataResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.29: Schema fragment for dstref:CreateResponse

Functions

lasso_idwsf2_dstref_create_response_new ()

```
LassoIdWsf2DstRefCreateResponse~*
lasso_idwsf2_dstref_create_response_new
                               (void);
```

Creates a new LassoIdWsf2DstRefCreateResponse object.

Returns

a newly created LassoIdWsf2DstRefCreateResponse object

Types and Values

struct LassoIdWsf2DstRefCreateResponse

```
struct LassoIdWsf2DstRefCreateResponse {
  LassoIdWsf2DstRefDataResponse parent;
};
```

11.31 LassoIdWsf2DstRefCreate

LassoIdWsf2DstRefCreate — <dstref:Create>

Functions

LassoIdWsf2DstRefCreate *	lasso_idwsf2_dstref_create_new ()

Types and Values

struct	LassoIdWsf2DstRefCreate

Description

```
<xs:complexType name="CreateType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="dstref:CreateItem" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="dstref:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.30: Schema fragment for dstref:Create

Functions

lasso_idwsf2_dstref_create_new ()

```
LassoIdWsf2DstRefCreate~*
lasso_idwsf2_dstref_create_new (void);
```

Creates a new LassoIdWsf2DstRefCreate object.

Returns

a newly created LassoIdWsf2DstRefCreate object

Types and Values

struct LassoIdWsf2DstRefCreate

```
struct LassoIdWsf2DstRefCreate {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *CreateItem; /* of LassoIdWsf2DstRefCreateItem */
  GList *ResultQuery; /* of LassoIdWsf2DstRefResultQuery */
};
```

11.32 LassoIdWsf2DstRefDataResponse

LassoIdWsf2DstRefDataResponse — <dstref:DataResponse>

Functions

LassoIdWsf2DstRefDataResponse *	lasso_idwsf2_dstref_data_response_new ()

Types and Values

struct	LassoIdWsf2DstRefDataResponse

Description

```
<xs:complexType name="DataResponseType">
  <xs:complexContent>
    <xs:extension base="dst:DataResponseBaseType">
      <xs:sequence>
        <xs:element ref="dstref:ItemData" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.31: Schema fragment for dstref:DataResponse

Functions

lasso_idwsf2_dstref_data_response_new ()

```
LassoIdWsf2DstRefDataResponse~*
lasso_idwsf2_dstref_data_response_new (void);
```

Creates a new LassoIdWsf2DstRefDataResponse object.

Returns

a newly created LassoIdWsf2DstRefDataResponse object

Types and Values

struct LassoIdWsf2DstRefDataResponse

```
struct LassoIdWsf2DstRefDataResponse {
  LassoIdWsf2DstDataResponseBase parent;

  /* elements */
  GList *ItemData; /* of LassoIdWsf2DstRefItemData */
};
```

11.33 LassoIdWsf2DstRefData

LassoIdWsf2DstRefData — <dstref:Data>

Functions

LassoIdWsf2DstRefData *	lasso_idwsf2_dstref_data_new ()

Types and Values

struct	LassoIdWsf2DstRefData

Description

```

<xs:complexType name="DataType">
  <xs:complexContent>
    <xs:extension base="dstref:ItemDataType">
      <xs:attributeGroup ref="dst:PaginationResponseAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.32: Schema fragment for dstref:Data

Functions

lasso_idwsf2_dstref_data_new ()

```
LassoIdWsf2DstRefData~*
lasso_idwsf2_dstref_data_new (void);
```

Creates a new LassoIdWsf2DstRefData object.

Returns

a newly created LassoIdWsf2DstRefData object

Types and Values

struct LassoIdWsf2DstRefData

```
struct LassoIdWsf2DstRefData {
  LassoIdWsf2DstRefItemData parent;

  /* attributes */
  int remaining;
  int nextOffset;
  char *setID;
};
```

11.34 LassoIdWsf2DstRefDeleteItem

LassoIdWsf2DstRefDeleteItem — <dstref:DeleteItem>

Functions

LassoIdWsf2DstRefDeleteItem *	lasso_idwsf2_dstref_delete_item_new ()

Types and Values

struct	LassoIdWsf2DstRefDeleteItem

Description

```
<xs:complexType name="DeleteItemType">
  <xs:complexContent>
    <xs:extension base="dst:DeleteItemBaseType">
      <xs:sequence>
        <xs:element ref="dstref:Select" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.33: Schema fragment for dstref:DeleteItem

Functions

lasso_idwsf2_dstref_delete_item_new ()

```
LassoIdWsf2DstRefDeleteItem~*
lasso_idwsf2_dstref_delete_item_new (void);
```

Creates a new LassoIdWsf2DstRefDeleteItem object.

Returns

a newly created LassoIdWsf2DstRefDeleteItem object

Types and Values

struct LassoIdWsf2DstRefDeleteItem

```
struct LassoIdWsf2DstRefDeleteItem {
  LassoIdWsf2DstDeleteItemBase parent;

  /* elements */
  char *Select;
};
```

11.35 LassoIdWsf2DstRefDeleteResponse

LassoIdWsf2DstRefDeleteResponse — <dstref:DeleteResponse>

Functions

LassoIdWsf2DstRefDeleteResponse *	lasso_idwsf2_dstref_delete_response_new ()

Types and Values

struct	LassoIdWsf2DstRefDeleteResponse

Description

```
<xs:complexType name="DeleteResponseType">
  <xs:complexContent>
    <xs:extension base="lu:ResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.34: Schema fragment for dstref:DeleteResponse

Functions

lasso_idwsf2_dstref_delete_response_new ()

```
LassoIdWsf2DstRefDeleteResponse~*
lasso_idwsf2_dstref_delete_response_new
                                (void);
```

Creates a new LassoIdWsf2DstRefDeleteResponse object.

Returns

a newly created LassoIdWsf2DstRefDeleteResponse object

Types and Values

struct LassoIdWsf2DstRefDeleteResponse

```
struct LassoIdWsf2DstRefDeleteResponse {
  LassoIdWsf2UtilResponse parent;
};
```

11.36 LassoIdWsf2DstRefDelete

LassoIdWsf2DstRefDelete — <dstref:Delete>

Functions

LassoIdWsf2DstRefDelete *	lasso_idwsf2_dstref_delete_new ()

Types and Values

struct	LassoIdWsf2DstRefDelete

Description

```
<xs:complexType name="DeleteType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="dstref:DeleteItem" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.35: Schema fragment for dstref:Delete

Functions

lasso_idwsf2_dstref_delete_new ()

```
LassoIdWsf2DstRefDelete~*
lasso_idwsf2_dstref_delete_new (void);
```

Creates a new LassoIdWsf2DstRefDelete object.

Returns

a newly created LassoIdWsf2DstRefDelete object

Types and Values

struct LassoIdWsf2DstRefDelete

```
struct LassoIdWsf2DstRefDelete {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *DeleteItem; /* of LassoIdWsf2DstRefDeleteItem */
};
```

11.37 LassoIdWsf2DstRefItemData

LassoIdWsf2DstRefItemData — <dstref:ItemData>

Functions

LassoIdWsf2DstRefItemData *	lasso_idwsf2_dstref_item_data_new ()

Types and Values

struct	LassoIdWsf2DstRefItemData

Description

```
<xs:complexType name="ItemDataType">
  <xs:complexContent>
    <xs:extension base="dstref:AppDataType">
      <xs:attributeGroup ref="dst:ItemDataAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.36: Schema fragment for dstref:ItemData

Functions

lasso_idwsf2_dstref_item_data_new ()

```
LassoIdWsf2DstRefItemData~*
lasso_idwsf2_dstref_item_data_new (void);
```

Creates a new LassoIdWsf2DstRefItemData object.

Returns

a newly created LassoIdWsf2DstRefItemData object

Types and Values

struct LassoIdWsf2DstRefItemData

```
struct LassoIdWsf2DstRefItemData {
  LassoIdWsf2DstRefAppData parent;

  /* attributes */
  char *itemIDRef;
  char *notSorted;
  char *changeFormat;
};
```

11.38 LassoIdWsf2DstRefModifyItem

LassoIdWsf2DstRefModifyItem — <dstref:ModifyItem>

Functions

LassoIdWsf2DstRefModifyItem *	lasso_idwsf2_dstref_modify_item_new ()
LassoIdWsf2DstRefModifyItem *	lasso_idwsf2_dstref_modify_item_new_full ()

Types and Values

struct	LassoIdWsf2DstRefModifyItem

Description

```
<xs:complexType name="ModifyItemType">
  <xs:sequence>
    <xs:element ref="dstref:Select" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="dstref:NewData" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="dst:ModifyItemAttributeGroup"/>
</xs:complexType>
```

Figure 11.37: Schema fragment for dstref:ModifyItem

Functions

lasso_idwsf2_dstref_modify_item_new ()

```
LassoIdWsf2DstRefModifyItem~*
lasso_idwsf2_dstref_modify_item_new (void);
```

Creates a new LassoIdWsf2DstRefModifyItem object.

Returns

a newly created LassoIdWsf2DstRefModifyItem object

lasso_idwsf2_dstref_modify_item_new_full ()

```
LassoIdWsf2DstRefModifyItem~*
lasso_idwsf2_dstref_modify_item_new_full
                               (const gchar *item_xpath,
                                const gchar *item_id,
                                xmlNode *new_data,
                                gboolean overrideAllowed);
```

Types and Values

struct LassoIdWsf2DstRefModifyItem

```
struct LassoIdWsf2DstRefModifyItem {
  LassoNode parent;

  /* elements */
  char *Select;
  LassoIdWsf2DstRefAppData *NewData;
  /* attributes */
  char *notChangedSince;
  gboolean overrideAllowed;
  char *id;
  char *itemID;
};
```

11.39 LassoIdWsf2DstRefModifyResponse

LassoIdWsf2DstRefModifyResponse — <dstref:ModifyResponse>

Functions

LassoIdWsf2DstRefModifyResponse *	lasso_idwsf2_dstref_modify_response_new ()

Types and Values

struct	LassoIdWsf2DstRefModifyResponse

Description

```
<xs:complexType name="ModifyResponseType">
  <xs:complexContent>
    <xs:extension base="dstref:DataResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.38: Schema fragment for dstref:ModifyResponse

Functions

lasso_idwsf2_dstref_modify_response_new ()

```
LassoIdWsf2DstRefModifyResponse~*
lasso_idwsf2_dstref_modify_response_new
                               (void);
```

Creates a new LassoIdWsf2DstRefModifyResponse object.

Returns

a newly created LassoIdWsf2DstRefModifyResponse object

Types and Values

struct LassoIdWsf2DstRefModifyResponse

```
struct LassoIdWsf2DstRefModifyResponse {
  LassoIdWsf2DstRefDataResponse parent;
};
```

11.40 LassoIdWsf2DstRefModify

LassoIdWsf2DstRefModify — <dstref:Modify>

Functions

LassoIdWsf2DstRefModify *	lasso_idwsf2_dstref_modify_new ()

Types and Values

struct	LassoIdWsf2DstRefModify

Description

```
<xs:complexType name="ModifyType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="dstref:ModifyItem" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="dstref:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.39: Schema fragment for dstref:Modify

Functions

lasso_idwsf2_dstref_modify_new ()

```
LassoIdWsf2DstRefModify~*
lasso_idwsf2_dstref_modify_new (void);
```

Creates a new LassoIdWsf2DstRefModify object.

Returns

a newly created LassoIdWsf2DstRefModify object

Types and Values

struct LassoIdWsf2DstRefModify

```
struct LassoIdWsf2DstRefModify {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *ModifyItem; /* of LassoIdWsf2DstRefModifyItem */
  GList *ResultQuery; /* of LassoIdWsf2DstRefResultQuery */
};
```

11.41 LassoIdWsf2DstRefQueryItem

LassoIdWsf2DstRefQueryItem — <dstref:QueryItem>

Functions

LassoIdWsf2DstRefQueryItem *	lasso_idwsf2_dstref_query_item_new ()
LassoIdWsf2DstRefQueryItem *	lasso_idwsf2_dstref_query_item_new_full ()

Types and Values

struct	LassoIdWsf2DstRefQueryItem

Description

```
<xs:complexType name="QueryItemType">
  <xs:complexContent>
    <xs:extension base="dstref:ResultQueryType">
      <xs:attributeGroup ref="dst:PaginationAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.40: Schema fragment for dstref:QueryItem

Functions

lasso_idwsf2_dstref_query_item_new ()

```
LassoIdWsf2DstRefQueryItem~*
lasso_idwsf2_dstref_query_item_new (void);
```

Creates a new LassoIdWsf2DstRefQueryItem object.

Returns

a newly created LassoIdWsf2DstRefQueryItem object

lasso_idwsf2_dstref_query_item_new_full ()

```
LassoIdWsf2DstRefQueryItem~*
lasso_idwsf2_dstref_query_item_new_full
                               (const gchar *item_xpath,
                                const gchar *item_id);
```

Types and Values

struct LassoIdWsf2DstRefQueryItem

```
struct LassoIdWsf2DstRefQueryItem {
  LassoIdWsf2DstRefResultQuery parent;

  /* attributes */
  int count;
  int offset;
  char *setID;
  char *setReq;
};
```

11.42 LassoIdWsf2DstRefQueryResponse

LassoIdWsf2DstRefQueryResponse — <dstref:QueryResponse>

Functions

LassoIdWsf2DstRefQueryResponse *	lasso_idwsf2_dstref_query_response_new ()

Types and Values

struct	LassoIdWsf2DstRefQueryResponse

Description

```

<xs:complexType name="QueryResponseType">
  <xs:complexContent>
    <xs:extension base="dst:DataResponseBaseType">
      <xs:sequence>
        <xs:element ref="dst:TestResult" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="dstref:Data" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.41: Schema fragment for dstref:QueryResponse

Functions

lasso_idwsf2_dstref_query_response_new ()

```
LassoIdWsf2DstRefQueryResponse~*
lasso_idwsf2_dstref_query_response_new
                               (void);
```

Creates a new LassoIdWsf2DstRefQueryResponse object.

Returns

a newly created LassoIdWsf2DstRefQueryResponse object

Types and Values

struct LassoIdWsf2DstRefQueryResponse

```
struct LassoIdWsf2DstRefQueryResponse {
  LassoIdWsf2DstDataResponseBase parent;

  /* elements */
  GList *TestResult; /* of LassoNode */
  GList *Data; /* of LassoIdWsf2DstRefData */
};
```

11.43 LassoIdWsf2DstRefQuery

LassoIdWsf2DstRefQuery — <dstref:Query>

Functions

LassoIdWsf2DstRefQuery *	lasso_idwsf2_dstref_query_new ()

Types and Values

struct	LassoIdWsf2DstRefQuery

Description

```
<xs:complexType name="QueryType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="dstref:TestItem" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="dstref:QueryItem" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.42: Schema fragment for dstref:Query

Functions

lasso_idwsf2_dstref_query_new ()

```
LassoIdWsf2DstRefQuery~*
lasso_idwsf2_dstref_query_new (void);
```

Creates a new LassoIdWsf2DstRefQuery object.

Returns

a newly created LassoIdWsf2DstRefQuery object

Types and Values

struct LassoIdWsf2DstRefQuery

```
struct LassoIdWsf2DstRefQuery {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *TestItem; /* of LassoIdWsf2DstRefTestItem */
  GList *QueryItem; /* of LassoIdWsf2DstRefQueryItem */
};
```

11.44 LassoIdWsf2DstRefResultQuery

LassoIdWsf2DstRefResultQuery — <dstref:ResultQuery>

Functions

LassoIdWsf2DstRefResultQuery *	lasso_idwsf2_dstref_result_query_new ()

Types and Values

struct	LassoIdWsf2DstRefResultQuery

Description

```
<xs:complexType name="ResultQueryType">
  <xs:complexContent>
    <xs:extension base="dst:ResultQueryBaseType">
      <xs:sequence>
        <xs:element ref="dstref:Select" minOccurs="0" maxOccurs="1"/>
        <xs:element name="Sort" minOccurs="0" maxOccurs="1" type="dstref:SortType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.43: Schema fragment for dstref:ResultQuery

Functions

lasso_idwsf2_dstref_result_query_new ()

```
LassoIdWsf2DstRefResultQuery~*
lasso_idwsf2_dstref_result_query_new (void);
```

Creates a new LassoIdWsf2DstRefResultQuery object.

Returns

a newly created LassoIdWsf2DstRefResultQuery object

Types and Values

struct LassoIdWsf2DstRefResultQuery

```
struct LassoIdWsf2DstRefResultQuery {
  LassoIdWsf2DstResultQueryBase parent;

  /* elements */
  char *Select;
  char *Sort;
  GHashTable *namespaces;
};
```

11.45 LassoIdWsf2DstRefTestItem

LassoIdWsf2DstRefTestItem — <dstref:TestItem>

Functions

LassoIdWsf2DstRefTestItem *	lasso_idwsf2_dstref_test_item_new ()

Types and Values

struct	LassoIdWsf2DstRefTestItem

Description

```
<xs:complexType name="TestItemType">
  <xs:complexContent>
    <xs:extension base="dst:TestItemBaseType">
      <xs:sequence>
        <xs:element name="TestOp" minOccurs="0" maxOccurs="1" type="dstref:TestOpType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.44: Schema fragment for dstref:TestItem

Functions

lasso_idwsf2_dstref_test_item_new ()

```
LassoIdWsf2DstRefTestItem~*
lasso_idwsf2_dstref_test_item_new (void);
```

Creates a new LassoIdWsf2DstRefTestItem object.

Returns

a newly created LassoIdWsf2DstRefTestItem object

Types and Values

struct LassoIdWsf2DstRefTestItem

```
struct LassoIdWsf2DstRefTestItem {
  LassoIdWsf2DstTestItemBase parent;

  /* elements */
  char *TestOp;
};
```

11.46 LassoIdWsf2DstRequest

LassoIdWsf2DstRequest — <dst:Request>

Functions

LassoIdWsf2DstRequest *	lasso_idwsf2_dst_request_new ()

Types and Values

struct	LassoIdWsf2DstRequest

Description

```
<xs:complexType name="RequestType">
  <xs:sequence>
    <xs:element ref="lu:Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="lu:itemID" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.45: Schema fragment for dst:Request

Functions

lasso_idwsf2_dst_request_new ()

```
LassoIdWsf2DstRequest~*
lasso_idwsf2_dst_request_new (void);
```

Creates a new LassoIdWsf2DstRequest object.

Returns

a newly created LassoIdWsf2DstRequest object

Types and Values

struct LassoIdWsf2DstRequest

```
struct LassoIdWsf2DstRequest {
  LassoNode parent;

  /* elements */
  GList *Extension; /* of LassoIdWsf2Utilextension */
  /* attributes */
  char *itemID;
  GHashTable *attributes;
};
```

11.47 LassoIdWsf2DstResultQueryBase

LassoIdWsf2DstResultQueryBase — <dst:ResultQueryBase>

Functions

LassoIdWsf2DstResultQueryBase *	lasso_idwsf2_dst_result_query_base_new ()

Types and Values

struct	LassoIdWsf2DstResultQueryBase

Description

```
<xs:complexType name="ResultQueryBaseType">
  <xs:sequence>
    <xs:element ref="dst:ChangeFormat" minOccurs="0" maxOccurs="2"/>
  </xs:sequence>
  <xs:attributeGroup ref="dst:selectQualif"/>
  <xs:attribute ref="lu:itemIDRef" use="optional"/>
  <xs:attribute name="contingency" use="optional" type="xs:boolean"/>
  <xs:attribute name="includeCommonAttributes" use="optional" type="xs:boolean"
          default="0"/>
  <xs:attribute name="changedSince" use="optional" type="xs:dateTime"/>
  <xs:attribute ref="lu:itemID" use="optional"/>
</xs:complexType>
```

Figure 11.46: Schema fragment for dst:ResultQueryBase

Functions

lasso_idwsf2_dst_result_query_base_new ()

```
LassoIdWsf2DstResultQueryBase~*
lasso_idwsf2_dst_result_query_base_new
                               (void);
```

Creates a new LassoIdWsf2DstResultQueryBase object.

Returns

a newly created LassoIdWsf2DstResultQueryBase object

Types and Values

struct LassoIdWsf2DstResultQueryBase

```
struct LassoIdWsf2DstResultQueryBase {
  LassoNode parent;

  /* elements */
  char *ChangeFormat;
  /* attributes */
  char *itemIDRef;
  gboolean contingency;
  gboolean includeCommonAttributes;
  char *changedSince;
  char *itemID;
  char *objectType;
  char *predefined;
};
```

11.48 LassoIdWsf2DstTestItemBase

LassoIdWsf2DstTestItemBase — <dst:TestItemBase>

Functions

LassoIdWsf2DstTestItemBase *	lasso_idwsf2_dst_test_item_base_new ()

Types and Values

struct	LassoIdWsf2DstTestItemBase

Description

```
<xs:complexType name="TestItemBaseType">
  <xs:attributeGroup ref="dst:selectQualif"/>
  <xs:attribute name="id" use="optional" type="xs:ID"/>
  <xs:attribute ref="lu:itemID" use="optional"/>
</xs:complexType>
```

Figure 11.47: Schema fragment for dst:TestItemBase

Functions

lasso_idwsf2_dst_test_item_base_new ()

```
LassoIdWsf2DstTestItemBase~*
lasso_idwsf2_dst_test_item_base_new (void);
```

Creates a new LassoIdWsf2DstTestItemBase object.

Returns

a newly created LassoIdWsf2DstTestItemBase object

Types and Values

struct LassoIdWsf2DstTestItemBase

```
struct LassoIdWsf2DstTestItemBase {
  LassoNode parent;

  /* attributes */
  char *id;
  char *itemID;
  char *objectType;
  char *predefined;
};
```

11.49 LassoIdWsf2ImsIdentityMappingRequest

LassoIdWsf2ImsIdentityMappingRequest — <ims:IdentityMappingRequest>

Functions

LassoIdWsf2ImsIdentityMappingRequest *	lasso_idwsf2_ims_identity_mapping_request_new ()

Types and Values

struct	LassoIdWsf2ImsIdentityMappingRequest

Description

```

<xs:complexType name="IdentityMappingRequestType">
  <xs:sequence>
    <xs:element ref="MappingInput" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.48: Schema fragment for ims:IdentityMappingRequest

Functions

lasso_idwsf2_ims_identity_mapping_request_new ()

```
LassoIdWsf2ImsIdentityMappingRequest~*
lasso_idwsf2_ims_identity_mapping_request_new
                                (void);
```

Creates a new LassoIdWsf2ImsIdentityMappingRequest object.

Returns

a newly created LassoIdWsf2ImsIdentityMappingRequest object

Types and Values

struct LassoIdWsf2ImsIdentityMappingRequest

```
struct LassoIdWsf2ImsIdentityMappingRequest {
  LassoNode parent;

  /* elements */
  GList *MappingInput; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

11.50 LassoIdWsf2ImsIdentityMappingResponse

LassoIdWsf2ImsIdentityMappingResponse — <ims:IdentityMappingResponse>

Functions

LassoIdWsf2ImsIdentityMappingResponse *	lasso_idwsf2_ims_identity_mapping_response_new ()

Types and Values

struct	LassoIdWsf2ImsIdentityMappingResponse

Description

```

<xs:complexType name="IdentityMappingResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status"/>
    <xs:element ref="MappingOutput" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.49: Schema fragment for ims:IdentityMappingResponse

Functions

lasso_idwsf2_ims_identity_mapping_response_new ()

```
LassoIdWsf2ImsIdentityMappingResponse~*
lasso_idwsf2_ims_identity_mapping_response_new
                                (void);
```

Creates a new LassoIdWsf2ImsIdentityMappingResponse object.

Returns

a newly created LassoIdWsf2ImsIdentityMappingResponse object

Types and Values

struct LassoIdWsf2ImsIdentityMappingResponse

```
struct LassoIdWsf2ImsIdentityMappingResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  GList *MappingOutput; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

11.51 LassoIdWsf2ImsMappingInput

LassoIdWsf2ImsMappingInput — <ims:MappingInput>

Functions

LassoIdWsf2ImsMappingInput *	lasso_idwsf2_ims_mapping_input_new ()

Types and Values

struct	LassoIdWsf2ImsMappingInput

Description

```
<xs:complexType name="MappingInputType">
  <xs:sequence>
    <xs:element ref="sec:TokenPolicy" minOccurs="0"/>
    <xs:element ref="sec:Token" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="reqID" type="lu:IDType" use="optional"/>
</xs:complexType>
```

Figure 11.50: Schema fragment for ims:MappingInput

Functions

lasso_idwsf2_ims_mapping_input_new ()

```
LassoIdWsf2ImsMappingInput~*
lasso_idwsf2_ims_mapping_input_new (void);
```

Creates a new LassoIdWsf2ImsMappingInput object.

Returns

a newly created LassoIdWsf2ImsMappingInput object

Types and Values

struct LassoIdWsf2ImsMappingInput

```
struct LassoIdWsf2ImsMappingInput {
  LassoNode parent;

  /* elements */
  /* XXX */ void *TokenPolicy;
  /* XXX */ void *Token;
  /* attributes */
  char *reqID;
};
```

11.52 LassoIdWsf2ImsMappingOutput

LassoIdWsf2ImsMappingOutput — <ims:MappingOutput>

Functions

LassoIdWsf2ImsMappingOutput *	lasso_idwsf2_ims_mapping_output_new ()

Types and Values

struct	LassoIdWsf2ImsMappingOutput

Description

```
<xs:complexType name="MappingOutputType">
  <xs:sequence>
    <xs:element ref="sec:Token"/>
  </xs:sequence>
  <xs:attribute name="reqRef" type="lu:IDReferenceType" use="optional"/>
</xs:complexType>
```

Figure 11.51: Schema fragment for ims:MappingOutput

Functions

lasso_idwsf2_ims_mapping_output_new ()

```
LassoIdWsf2ImsMappingOutput~*
lasso_idwsf2_ims_mapping_output_new (void);
```

Creates a new LassoIdWsf2ImsMappingOutput object.

Returns

a newly created LassoIdWsf2ImsMappingOutput object

Types and Values

struct LassoIdWsf2ImsMappingOutput

```
struct LassoIdWsf2ImsMappingOutput {
  LassoNode parent;

  /* elements */
  /* XXX */ void *Token;
  /* attributes */
  char *reqRef;
};
```

11.53 LassoIsHelp

LassoIsHelp — <is:Help>

Functions

#define	LASSO_IS_IS_HELP()
LassoIsHelp *	lasso_is_help_new ()

Types and Values

struct	LassoIsHelp

Description

```
<xs:complexType name="HelpType">
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="link" type="xs:anyURI" use="optional"/>
  <xs:attribute name="moreLink" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

Figure 11.52: Schema fragment for is:Help

Functions

LASSO_IS_IS_HELP()

```
#define LASSO_IS_IS_HELP(obj) (G_TYPE_CHECK_INSTANCE_TYPE((obj), LASSO_TYPE_IS_HELP))
```

lasso_is_help_new ()

```
LassoIsHelp~*
lasso_is_help_new (void);
```

Types and Values

struct LassoIsHelp

```
struct LassoIsHelp {
  LassoNode parent;

  char *label;
  char *link;
  char *moreLink;
};
```

11.54 LassoIsInquiryElement

LassoIsInquiryElement — <is:InquiryElement>

Functions

#define	LASSO_IS_IS_INQUIRY_ELEMENT()
LassoIsInquiryElement *	lasso_is_inquiry_element_new ()

Types and Values

struct	LassoIsInquiryElement

Description

```
<xs:complexType name="InquiryElementType" abstract="true">
  <xs:sequence>
    <xs:element ref="Help" minOccurs="0"/>
    <xs:element ref="Hint" minOccurs="0"/>
    <xs:element name="Label" type="xs:normalizedString" minOccurs="0"/>
    <xs:element name="Value" type="xs:normalizedString" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:ID" use="required"/>
</xs:complexType>
```

Figure 11.53: Schema fragment for is:InquiryElement

Functions

LASSO_IS_IS_INQUIRY_ELEMENT()

```
#define             LASSO_IS_IS_INQUIRY_ELEMENT(obj)
```

lasso_is_inquiry_element_new ()

```
LassoIsInquiryElement~*
lasso_is_inquiry_element_new ();
```

Types and Values

struct LassoIsInquiryElement

```
struct LassoIsInquiryElement {
  LassoNode parent;

  LassoIsHelp *Help;
  char *Hint;
  char *Label;
  char *Value;

  char *name;
};
```

11.55 LassoIsInquiry

LassoIsInquiry — <is:Inquiry>

Functions

#define	LASSO_IS_IS_INQUIRY()
LassoIsInquiry *	lasso_is_inquiry_new ()

Types and Values

struct	LassoIsInquiry

Description

```
<xs:complexType name="InquiryType">
  <xs:sequence>
    <xs:element ref="Help" minOccurs="0"/>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Select" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Confirm" type="InquiryElementType"
        minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="Text" minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
    <xs:attribute name="title" type="xs:string" use="optional"/>
  </xs:complexType>
```

Figure 11.54: Schema fragment for is:Inquiry

Functions

LASSO_IS_IS_INQUIRY()

```
#define LASSO_IS_IS_INQUIRY(obj) (G_TYPE_CHECK_INSTANCE_TYPE((obj), LASSO_TYPE_IS_INQUIRY))
```

lasso_is_inquiry_new ()

```
LassoIsInquiry~*
lasso_is_inquiry_new (void);
```

Types and Values

struct LassoIsInquiry

```
struct LassoIsInquiry {
  LassoNode parent;

  GList *Help; /* of LassoNode */
  GList *Select; /* of LassoNode */
  GList *Confirm; /* of LassoNode */
  GList *Text; /* of LassoNode */

  char *id;
  char *title;
};
```

11.56 LassoIsInteractionRequest

LassoIsInteractionRequest — <is:InteractionRequest>

Functions

#define	LASSO_IS_IS_INTERACTION_REQUEST()
LassoIsInteractionRequest *	lasso_is_interaction_request_new ()

Types and Values

struct	LassoIsInteractionRequest

Description

```
<xs:complexType name="InteractionRequestType">
  <xs:sequence>
    <xs:element ref="Inquiry" maxOccurs="unbounded"/>
    <xs:element ref="ds:KeyInfo" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="language" type="xs:NMTOKENS" use="optional"/>
  <xs:attribute name="maxInteractTime" type="xs:integer" use="optional"/>
  <xs:attribute name="signed" type="xs:token" use="optional"/>
</xs:complexType>
```

Figure 11.55: Schema fragment for is:InteractionRequest

Functions

LASSO_IS_IS_INTERACTION_REQUEST()

```
#define             LASSO_IS_IS_INTERACTION_REQUEST(obj)
```

lasso_is_interaction_request_new ()

```
LassoIsInteractionRequest~*
lasso_is_interaction_request_new (void);
```

Types and Values

struct LassoIsInteractionRequest

```
struct LassoIsInteractionRequest {
  LassoNode parent;

  LassoDiscoResourceID *ResourceID;
  LassoDiscoEncryptedResourceID *EncryptedResourceID;
  GList *Inquiry; /* of LassoNode */

  char *id;
  char *language;
  int maxInteractTime;
  LassoDsKeyInfo *KeyInfo;
  char *signed_attribute;
};
```

11.57 LassoIsInteractionResponse

LassoIsInteractionResponse — <is:InteractionResponse>

Functions

#define	LASSO_IS_IS_INTERACTION_RESPONSE()
LassoIsInteractionResponse *	lasso_is_interaction_response_new ()

Types and Values

struct	LassoIsInteractionResponse

Description

```
<xs:complexType name="InteractionResponseType">
  <xs:sequence>
    <xs:element ref="lu:Status"/>
    <xs:choice>
      <xs:element name="InteractionStatement" type="InteractionStatementType"
        minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Parameter" type="ParameterType" minOccurs="0" maxOccurs=" ↩
            unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
```

Figure 11.56: Schema fragment for is:InteractionResponse

Functions

LASSO_IS_IS_INTERACTION_RESPONSE()

```
#define             LASSO_IS_IS_INTERACTION_RESPONSE(obj)
```

lasso_is_interaction_response_new ()

```
LassoIsInteractionResponse~*
lasso_is_interaction_response_new ();
```

Types and Values

struct LassoIsInteractionResponse

```
struct LassoIsInteractionResponse {
  LassoNode parent; /* FIXME : inherit of LassoIsInquiryElement */

  LassoUtilityStatus *Status;
  GList *InteractionStatement; /* of LassoNode */
  GList *Parameter; /* of LassoNode */
};
```

11.58 LassoIsInteractionStatement

LassoIsInteractionStatement — <is:InteractionStatement>

Functions

#define	LASSO_IS_IS_INTERACTION_STATEMENT()
LassoIsInteractionStatement *	lasso_is_interaction_statement_new ()

Types and Values

struct	LassoIsInteractionStatement

Description

```

<xs:complexType name="InteractionStatementType">
  <xs:sequence>
    <xs:element ref="Inquiry" maxOccurs="unbounded"/>
    <xs:element ref="ds:Signature"/>
  </xs:sequence>
</xs:complexType>
```

Figure 11.57: Schema fragment for is:InteractionStatement

Functions

LASSO_IS_IS_INTERACTION_STATEMENT()

```
#define             LASSO_IS_IS_INTERACTION_STATEMENT(obj)
```

lasso_is_interaction_statement_new ()

```
LassoIsInteractionStatement~*
lasso_is_interaction_statement_new ();
```

Types and Values

struct LassoIsInteractionStatement

```
struct LassoIsInteractionStatement {
  LassoNode parent; /* FIXME : inherit of LassoIsInquiryElement */

  LassoIsInquiry *Inquiry;
};
```

11.59 LassoIsItem

LassoIsItem — <is:Item>

Functions

#define	LASSO_IS_IS_ITEM()
LassoIsItem *	lasso_is_item_new ()

Types and Values

struct	LassoIsItem

Description

```
<xs:element name="Item" minOccurs="2" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Hint" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="label" type="xs:string" use="optional"/>
    <xs:attribute name="value" type="xs:NMTOKEN" use="required"/>
  </xs:complexType>
</xs:element>
```

Figure 11.58: Schema fragment for is:Item

Functions

LASSO_IS_IS_ITEM()

```
#define LASSO_IS_IS_ITEM(obj) (G_TYPE_CHECK_INSTANCE_TYPE((obj), LASSO_TYPE_IS_ITEM))
```

lasso_is_item_new ()

```
LassoIsItem~*
lasso_is_item_new (const char *value);
```

Types and Values

struct LassoIsItem

```
struct LassoIsItem {
  LassoNode parent;

  char *Hint;
  char *label;
  char *value;
};
```

11.60 LassoIsParameter

LassoIsParameter — <is:Parameter>

Functions

#define	LASSO_IS_IS_PARAMETER()
LassoIsParameter *	lasso_is_parameter_new ()

Types and Values

struct	LassoIsParameter

Description

```
<xs:complexType name="ParameterType">
  <xs:attribute name="name" type="xs:ID" use="required"/>
  <xs:attribute name="value" type="xs:string" use="required"/>
</xs:complexType>
```

Figure 11.59: Schema fragment for is:Parameter

Functions

LASSO_IS_IS_PARAMETER()

```
#define LASSO_IS_IS_PARAMETER(obj) (G_TYPE_CHECK_INSTANCE_TYPE((obj),  ↩
    LASSO_TYPE_IS_PARAMETER))
```

lasso_is_parameter_new ()

```
LassoIsParameter~*
lasso_is_parameter_new ();
```

Types and Values

struct LassoIsParameter

```
struct LassoIsParameter {
  LassoNode parent;

  char *name;
  char *value;
};
```

11.61 LassoIsSelect

LassoIsSelect — <is:Select>

Functions

#define	LASSO_IS_IS_SELECT()
LassoIsSelect *	lasso_is_select_new ()

Types and Values

struct	LassoIsSelect

Description

```
<xs:complexType name="SelectType">
  <xs:complexContent>
    <xs:extension base="InquiryElementType">
      <xs:sequence>
        <xs:element name="Item" minOccurs="2" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="Hint" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="label" type="xs:string" use="optional"/>
            <xs:attribute name="value" type="xs:NMTOKEN" use="required"/>
          </xs:complexType>
```

Figure 11.60: Schema fragment for is:Select

Functions

LASSO_IS_IS_SELECT()

```
#define LASSO_IS_IS_SELECT(obj) (G_TYPE_CHECK_INSTANCE_TYPE((obj), LASSO_TYPE_IS_SELECT))
```

lasso_is_select_new ()

```
LassoIsSelect~*
lasso_is_select_new (void);
```

Types and Values

struct LassoIsSelect

```
struct LassoIsSelect {
  LassoNode parent; /* FIXME : must inherit of InquiryElement class */

  GList *Item; /* of LassoNode */

  gboolean multiple;
};
```

11.62 LassoIsText

LassoIsText — <is:Text>

Functions

#define	LASSO_IS_IS_TEXT()
LassoIsText *	lasso_is_text_new ()

Types and Values

struct	LassoIsText

Description

```
<xs:complexType name="TextType">
  <xs:complexContent>
    <xs:extension base="InquiryElementType">
      <xs:attribute name="minChars" type="xs:integer" use="optional"/>
      <xs:attribute name="maxChars" type="xs:integer" use="optional"/>
      <xs:attribute name="format" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.61: Schema fragment for is:Text

Functions

LASSO_IS_IS_TEXT()

```
#define LASSO_IS_IS_TEXT(obj) (G_TYPE_CHECK_INSTANCE_TYPE((obj), LASSO_TYPE_IS_TEXT))
```

lasso_is_text_new ()

```
LassoIsText~*
lasso_is_text_new ();
```

Types and Values

struct LassoIsText

```
struct LassoIsText {
  LassoNode parent; /* FIXME : inherit of LassoIsInquiryElement */

  int minChars;
  int maxChars;
  char *format;
};
```

11.63 LassoIdWsf2PsAddCollectionRequest

LassoIdWsf2PsAddCollectionRequest — <ps:AddCollectionRequest>

Functions

LassoIdWsf2PsAddCollectionRequest *	lasso_idwsf2_ps_add_collection_request_new ()

Types and Values

struct	LassoIdWsf2PsAddCollectionRequest

Description

```
<xs:complexType name="AddCollectionRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Object"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.62: Schema fragment for ps:AddCollectionRequest

Functions

lasso_idwsf2_ps_add_collection_request_new ()

```
LassoIdWsf2PsAddCollectionRequest~*
lasso_idwsf2_ps_add_collection_request_new
                               (void);
```

Creates a new LassoIdWsf2PsAddCollectionRequest object.

Returns

a newly created LassoIdWsf2PsAddCollectionRequest object

Types and Values

struct LassoIdWsf2PsAddCollectionRequest

```
struct LassoIdWsf2PsAddCollectionRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *Object;
  /* XXX */ void *Subscription;
};
```

11.64 LassoIdWsf2PsAddCollectionResponse

LassoIdWsf2PsAddCollectionResponse — <ps:AddCollectionResponse>

Functions

LassoIdWsf2PsAddCollectionResponse *	lasso_idwsf2_ps_add_collection_response_new ()

Types and Values

struct	LassoIdWsf2PsAddCollectionResponse

Description

```
<xs:complexType name="AddCollectionResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Object" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.63: Schema fragment for ps:AddCollectionResponse

Functions

lasso_idwsf2_ps_add_collection_response_new ()

```
LassoIdWsf2PsAddCollectionResponse~*
lasso_idwsf2_ps_add_collection_response_new
                                (void);
```

Creates a new LassoIdWsf2PsAddCollectionResponse object.

Returns

a newly created LassoIdWsf2PsAddCollectionResponse object

Types and Values

struct LassoIdWsf2PsAddCollectionResponse

```
struct LassoIdWsf2PsAddCollectionResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  /* XXX */ void *Object;
};
```

11.65 LassoIdWsf2PsAddEntityRequest

LassoIdWsf2PsAddEntityRequest — <ps:AddEntityRequest>

Functions

LassoIdWsf2PsAddEntityRequest *	lasso_idwsf2_ps_add_entity_request_new ()

Types and Values

struct	LassoIdWsf2PsAddEntityRequest

Description

```
<xs:complexType name="AddEntityRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Object"/>
        <xs:element ref="PStoSPRedirectURL" minOccurs="0"/>
        <xs:element ref="CreatePSObject" minOccurs="0"/>
        <xs:element ref="Subscription" minOccurs="0"/>
        <xs:element ref="sec:TokenPolicy" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.64: Schema fragment for ps:AddEntityRequest

Functions

lasso_idwsf2_ps_add_entity_request_new ()

```
LassoIdWsf2PsAddEntityRequest~*
lasso_idwsf2_ps_add_entity_request_new
                                (void);
```

Creates a new LassoIdWsf2PsAddEntityRequest object.

Returns

a newly created LassoIdWsf2PsAddEntityRequest object

Types and Values

struct LassoIdWsf2PsAddEntityRequest

```
struct LassoIdWsf2PsAddEntityRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *Object;
  /* XXX */ void *PStoSPRedirectURL;
  char *CreatePSObject;
  /* XXX */ void *Subscription;
  /* XXX */ void *TokenPolicy;
};
```

11.66 LassoIdWsf2PsAddEntityResponse

LassoIdWsf2PsAddEntityResponse — <ps:AddEntityResponse>

Functions

LassoIdWsf2PsAddEntityResponse *	lasso_idwsf2_ps_add_entity_response_new ()

Types and Values

struct	LassoIdWsf2PsAddEntityResponse

Description

```
<xs:complexType name="AddEntityResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Object" minOccurs="0"/>
        <xs:element ref="SPtoPSRedirectURL" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="QueryString" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.65: Schema fragment for ps:AddEntityResponse

Functions

lasso_idwsf2_ps_add_entity_response_new ()

```
LassoIdWsf2PsAddEntityResponse~*
lasso_idwsf2_ps_add_entity_response_new
                                (void);
```

Creates a new LassoIdWsf2PsAddEntityResponse object.

Returns

a newly created LassoIdWsf2PsAddEntityResponse object

Types and Values

struct LassoIdWsf2PsAddEntityResponse

```
struct LassoIdWsf2PsAddEntityResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  /* XXX */ void *Object;
  /* XXX */ void *SPtoPSRedirectURL;
  /* XXX */ void *QueryString;
};
```

11.67 LassoIdWsf2PsAddKnownEntityRequest

LassoIdWsf2PsAddKnownEntityRequest — <ps:AddKnownEntityRequest>

Functions

LassoIdWsf2PsAddKnownEntityRequest *	lasso_idwsf2_ps_add_known_entity_request_new ()

Types and Values

struct	LassoIdWsf2PsAddKnownEntityRequest

Description

```
<xs:complexType name="AddKnownEntityRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Object"/>
        <xs:element ref="sec:Token"/>
        <xs:element ref="CreatePSObject" minOccurs="0"/>
        <xs:element ref="Subscription" minOccurs="0"/>
        <xs:element ref="sec:TokenPolicy" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.66: Schema fragment for ps:AddKnownEntityRequest

Functions

lasso_idwsf2_ps_add_known_entity_request_new ()

```
LassoIdWsf2PsAddKnownEntityRequest~*
lasso_idwsf2_ps_add_known_entity_request_new
                               (void);
```

Creates a new LassoIdWsf2PsAddKnownEntityRequest object.

Returns

a newly created LassoIdWsf2PsAddKnownEntityRequest object

Types and Values

struct LassoIdWsf2PsAddKnownEntityRequest

```
struct LassoIdWsf2PsAddKnownEntityRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *Object;
  /* XXX */ void *Token;
  char *CreatePSObject;
  /* XXX */ void *Subscription;
  /* XXX */ void *TokenPolicy;
};
```

11.68 LassoIdWsf2PsAddKnownEntityResponse

LassoIdWsf2PsAddKnownEntityResponse — <ps:AddKnownEntityResponse>

Functions

LassoIdWsf2PsAddKnownEntityResponse *	lasso_idwsf2_ps_add_known_entity_response_new ()

Types and Values

struct	LassoIdWsf2PsAddKnownEntityResponse

Description

```
<xs:complexType name="AddKnownEntityResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Object" minOccurs="0"/>
        <xs:element ref="SPtoPSRedirectURL" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="QueryString" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.67: Schema fragment for ps:AddKnownEntityResponse

Functions

lasso_idwsf2_ps_add_known_entity_response_new ()

```
LassoIdWsf2PsAddKnownEntityResponse~*
lasso_idwsf2_ps_add_known_entity_response_new
                                (void);
```

Creates a new LassoIdWsf2PsAddKnownEntityResponse object.

Returns

a newly created LassoIdWsf2PsAddKnownEntityResponse object

Types and Values

struct LassoIdWsf2PsAddKnownEntityResponse

```
struct LassoIdWsf2PsAddKnownEntityResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  /* XXX */ void *Object;
  /* XXX */ void *SPtoPSRedirectURL;
  /* XXX */ void *QueryString;
};
```

11.69 LassoIdWsf2PsAddToCollectionRequest

LassoIdWsf2PsAddToCollectionRequest — <ps:AddToCollectionRequest>

Functions

LassoIdWsf2PsAddToCollectionRequest *	lasso_idwsf2_ps_add_to_collection_request_new ()

Types and Values

struct	LassoIdWsf2PsAddToCollectionRequest

Description

```
<xs:complexType name="AddToCollectionRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID"/>
        <xs:element ref="ObjectID" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.68: Schema fragment for ps:AddToCollectionRequest

Functions

lasso_idwsf2_ps_add_to_collection_request_new ()

```
LassoIdWsf2PsAddToCollectionRequest~*
lasso_idwsf2_ps_add_to_collection_request_new
                                (void);
```

Creates a new LassoIdWsf2PsAddToCollectionRequest object.

Returns

a newly created LassoIdWsf2PsAddToCollectionRequest object

Types and Values

struct LassoIdWsf2PsAddToCollectionRequest

```
struct LassoIdWsf2PsAddToCollectionRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *TargetObjectID;
  GList *ObjectID; /* of LassoNode */
  /* XXX */ void *Subscription;
};
```

11.70 LassoIdWsf2PsGetObjectInfoRequest

LassoIdWsf2PsGetObjectInfoRequest — <ps:GetObjectInfoRequest>

Functions

LassoIdWsf2PsGetObjectInfoRequest *	lasso_idwsf2_ps_get_object_info_request_new ()

Types and Values

struct	LassoIdWsf2PsGetObjectInfoRequest

Description

```
<xs:complexType name="GetObjectInfoRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID" minOccurs="0"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.69: Schema fragment for ps:GetObjectInfoRequest

Functions

lasso_idwsf2_ps_get_object_info_request_new ()

```
LassoIdWsf2PsGetObjectInfoRequest~*
lasso_idwsf2_ps_get_object_info_request_new
                               (void);
```

Creates a new LassoIdWsf2PsGetObjectInfoRequest object.

Returns

a newly created LassoIdWsf2PsGetObjectInfoRequest object

Types and Values

struct LassoIdWsf2PsGetObjectInfoRequest

```
struct LassoIdWsf2PsGetObjectInfoRequest {
  LassoIdWsf2PsRequestAbstract parent;
  /* elements */
  /* XXX */ void *TargetObjectID;
  /* XXX */ void *Subscription;
};
```

11.71 LassoIdWsf2PsGetObjectInfoResponse

LassoIdWsf2PsGetObjectInfoResponse — <ps:GetObjectInfoResponse>

Functions

LassoIdWsf2PsGetObjectInfoResponse *	lasso_idwsf2_ps_get_object_info_response_new ()

Types and Values

struct	LassoIdWsf2PsGetObjectInfoResponse

Description

```
<xs:complexType name="GetObjectInfoResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Object" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.70: Schema fragment for ps:GetObjectInfoResponse

Functions

lasso_idwsf2_ps_get_object_info_response_new ()

```
LassoIdWsf2PsGetObjectInfoResponse~*
lasso_idwsf2_ps_get_object_info_response_new
                                (void);
```

Creates a new LassoIdWsf2PsGetObjectInfoResponse object.

Returns

a newly created LassoIdWsf2PsGetObjectInfoResponse object

Types and Values

struct LassoIdWsf2PsGetObjectInfoResponse

```
struct LassoIdWsf2PsGetObjectInfoResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  /* XXX */ void *Object;
};
```

11.72 LassoIdWsf2PsItemData

LassoIdWsf2PsItemData — <ps:ItemData>

Functions

LassoIdWsf2PsItemData *	lasso_idwsf2_ps_item_data_new ()

Types and Values

struct	LassoIdWsf2PsItemData

Description

```

<xs:complexType name="ItemDataType">
  <xs:sequence>
    <xs:element ref="Object"/>
  </xs:sequence>
</xs:complexType>
```

Figure 11.71: Schema fragment for ps:ItemData

Functions

lasso_idwsf2_ps_item_data_new ()

```
LassoIdWsf2PsItemData~*
lasso_idwsf2_ps_item_data_new (void);
```

Creates a new LassoIdWsf2PsItemData object.

Returns

a newly created LassoIdWsf2PsItemData object

Types and Values

struct LassoIdWsf2PsItemData

```
struct LassoIdWsf2PsItemData {
  LassoNode parent;

  /* elements */
  /* XXX */ void *Object;
};
```

11.73 LassoIdWsf2PsListMembersRequest

LassoIdWsf2PsListMembersRequest — <ps:ListMembersRequest>

Functions

LassoIdWsf2PsListMembersRequest *	lasso_idwsf2_ps_list_members_request_new ()

Types and Values

struct	LassoIdWsf2PsListMembersRequest

Description

```
<xs:complexType name="ListMembersRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID" minOccurs="0"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="Structured" type="xs:anyURI" use="optional"/>
      <xs:attribute name="Count" type="xs:nonNegativeInteger" use="optional"/>
      <xs:attribute name="Offset" type="xs:nonNegativeInteger" default="0" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.72: Schema fragment for ps:ListMembersRequest

Functions

lasso_idwsf2_ps_list_members_request_new ()

```
LassoIdWsf2PsListMembersRequest~*
lasso_idwsf2_ps_list_members_request_new
                                (void);
```

Creates a new LassoIdWsf2PsListMembersRequest object.

Returns

a newly created LassoIdWsf2PsListMembersRequest object

Types and Values

struct LassoIdWsf2PsListMembersRequest

```
struct LassoIdWsf2PsListMembersRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *TargetObjectID;
  /* XXX */ void *Subscription;
  /* attributes */
  char *Structured;
  int Count;
  int Offset;
};
```

11.74 LassoIdWsf2PsListMembersResponse

LassoIdWsf2PsListMembersResponse — <ps:ListMembersResponse>

Functions

LassoIdWsf2PsListMembersResponse *	lasso_idwsf2_ps_list_members_response_new ()

Types and Values

struct	LassoIdWsf2PsListMembersResponse

Description

```
<xs:complexType name="ListMembersResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Object" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.73: Schema fragment for ps:ListMembersResponse

Functions

lasso_idwsf2_ps_list_members_response_new ()

```
LassoIdWsf2PsListMembersResponse~*
lasso_idwsf2_ps_list_members_response_new
                                (void);
```

Creates a new LassoIdWsf2PsListMembersResponse object.

Returns

a newly created LassoIdWsf2PsListMembersResponse object

Types and Values

struct LassoIdWsf2PsListMembersResponse

```
struct LassoIdWsf2PsListMembersResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  GList *Object; /* of LassoNode */
};
```

11.75 LassoIdWsf2PsNotification

LassoIdWsf2PsNotification — <ps:Notification>

Functions

LassoIdWsf2PsNotification *	lasso_idwsf2_ps_notification_new ()

Types and Values

struct	LassoIdWsf2PsNotification

Description

```
<xs:complexType name="NotificationType">
  <xs:complexContent>
    <xs:extension base="subs:NotificationType">
      <xs:sequence>
        <xs:element ref="ItemData" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.74: Schema fragment for ps:Notification

Functions

lasso_idwsf2_ps_notification_new ()

```
LassoIdWsf2PsNotification~*
lasso_idwsf2_ps_notification_new (void);
```

Creates a new LassoIdWsf2PsNotification object.

Returns

a newly created LassoIdWsf2PsNotification object

Types and Values

struct LassoIdWsf2PsNotification

```
struct LassoIdWsf2PsNotification {
  LassoIdWsf2SubsNotification parent;

  /* elements */
  GList *ItemData; /* of LassoNode */
};
```

11.76 LassoIdWsf2PsNotify

LassoIdWsf2PsNotify — <ps:Notify>

Functions

LassoIdWsf2PsNotify *	lasso_idwsf2_ps_notify_new ()

Types and Values

struct	LassoIdWsf2PsNotify

Description

```
<xs:complexType name="NotifyType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Notification" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="subs:NotifyAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.75: Schema fragment for ps:Notify

Functions

lasso_idwsf2_ps_notify_new ()

```
LassoIdWsf2PsNotify~*
lasso_idwsf2_ps_notify_new (void);
```

Creates a new LassoIdWsf2PsNotify object.

Returns

a newly created LassoIdWsf2PsNotify object

Types and Values

struct LassoIdWsf2PsNotify

```
struct LassoIdWsf2PsNotify {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  GList *Notification; /* of LassoNode */
  /* attributes */
  char *timeStamp;
};
```

11.77 LassoIdWsf2PsObject

LassoIdWsf2PsObject — <ps:Object>

Functions

LassoIdWsf2PsObject *	lasso_idwsf2_ps_object_new ()

Types and Values

struct	LassoIdWsf2PsObject

Description

```
<xs:complexType name="ObjectType">
  <xs:sequence>
    <xs:element ref="ObjectID" minOccurs="0"/>
    <xs:element name="DisplayName" type="LocalizedDisplayNameType" minOccurs="1"
            maxOccurs="unbounded"/>
    <xs:element name="Tag" type="TagType" minOccurs="0"/>
    <xs:element ref="Object" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ObjectRef" type="ObjectIDType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="NodeType" type="xs:anyURI" use="required"/>
  <xs:attribute name="CreatedDateTime" type="xs:dateTime" use="optional"/>
  <xs:attribute name="ModifiedDateTime" type="xs:dateTime" use="optional"/>
</xs:complexType>
```

Figure 11.76: Schema fragment for ps:Object

Functions

lasso_idwsf2_ps_object_new ()

```
LassoIdWsf2PsObject~*
lasso_idwsf2_ps_object_new (void);
```

Creates a new LassoIdWsf2PsObject object.

Returns

a newly created LassoIdWsf2PsObject object

Types and Values

struct LassoIdWsf2PsObject

```
struct LassoIdWsf2PsObject {
  LassoNode parent;

  /* elements */
  /* XXX */ void *ObjectID;
  GList *DisplayName; /* of LassoNode */
  /* XXX */ void *Tag;
  GList *Object; /* of LassoNode */
  GList *ObjectRef; /* of LassoNode */
  /* attributes */
  char *NodeType;
  char *CreatedDateTime;
  char *ModifiedDateTime;
};
```

11.78 LassoIdWsf2PsQueryObjectsRequest

LassoIdWsf2PsQueryObjectsRequest — <ps:QueryObjectsRequest>

Functions

LassoIdWsf2PsQueryObjectsRequest *	lasso_idwsf2_ps_query_objects_request_new ()

Types and Values

struct	LassoIdWsf2PsQueryObjectsRequest

Description

```
<xs:complexType name="QueryObjectsRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element name="Filter" type="xs:string"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="Count" type="xs:nonNegativeInteger" use="optional"/>
      <xs:attribute name="Offset" type="xs:nonNegativeInteger" default="0" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.77: Schema fragment for ps:QueryObjectsRequest

Functions

lasso_idwsf2_ps_query_objects_request_new ()

```
LassoIdWsf2PsQueryObjectsRequest~*
lasso_idwsf2_ps_query_objects_request_new
                                (void);
```

Creates a new LassoIdWsf2PsQueryObjectsRequest object.

Returns

a newly created LassoIdWsf2PsQueryObjectsRequest object

Types and Values

struct LassoIdWsf2PsQueryObjectsRequest

```
struct LassoIdWsf2PsQueryObjectsRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  char *Filter;
  /* XXX */ void *Subscription;
  /* attributes */
  int Count;
  int Offset;
};
```

11.79 LassoIdWsf2PsQueryObjectsResponse

LassoIdWsf2PsQueryObjectsResponse — <ps:QueryObjectsResponse>

Functions

LassoIdWsf2PsQueryObjectsResponse	*lasso_idwsf2_ps_query_objects_response_new ()

Types and Values

struct	LassoIdWsf2PsQueryObjectsResponse

Description

```
<xs:complexType name="QueryObjectsResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="Object" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.78: Schema fragment for ps:QueryObjectsResponse

Functions

lasso_idwsf2_ps_query_objects_response_new ()

```
LassoIdWsf2PsQueryObjectsResponse~*
lasso_idwsf2_ps_query_objects_response_new
                                (void);
```

Creates a new LassoIdWsf2PsQueryObjectsResponse object.

Returns

a newly created LassoIdWsf2PsQueryObjectsResponse object

Types and Values

struct LassoIdWsf2PsQueryObjectsResponse

```
struct LassoIdWsf2PsQueryObjectsResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  GList *Object; /* of LassoNode */
};
```

11.80 LassoIdWsf2PsRemoveCollectionRequest

LassoIdWsf2PsRemoveCollectionRequest — <ps:RemoveCollectionRequest>

Functions

LassoIdWsf2PsRemoveCollectionRequest *	lasso_idwsf2_ps_remove_collection_request_new ()

Types and Values

struct	LassoIdWsf2PsRemoveCollectionRequest

Description

```
<xs:complexType name="RemoveCollectionRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.79: Schema fragment for ps:RemoveCollectionRequest

Functions

lasso_idwsf2_ps_remove_collection_request_new ()

```
LassoIdWsf2PsRemoveCollectionRequest~*
lasso_idwsf2_ps_remove_collection_request_new
                                (void);
```

Creates a new LassoIdWsf2PsRemoveCollectionRequest object.

Returns

a newly created LassoIdWsf2PsRemoveCollectionRequest object

Types and Values

struct LassoIdWsf2PsRemoveCollectionRequest

```
struct LassoIdWsf2PsRemoveCollectionRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  GList *TargetObjectID; /* of LassoNode */
};
```

11.81 LassoIdWsf2PsRemoveEntityRequest

LassoIdWsf2PsRemoveEntityRequest — <ps:RemoveEntityRequest>

Functions

LassoIdWsf2PsRemoveEntityRequest *	lasso_idwsf2_ps_remove_entity_request_new ()

Types and Values

struct	LassoIdWsf2PsRemoveEntityRequest

Description

```

<xs:complexType name="RemoveEntityRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.80: Schema fragment for ps:RemoveEntityRequest

Functions

lasso_idwsf2_ps_remove_entity_request_new ()

```
LassoIdWsf2PsRemoveEntityRequest~*
lasso_idwsf2_ps_remove_entity_request_new
                                (void);
```

Creates a new LassoIdWsf2PsRemoveEntityRequest object.

Returns

a newly created LassoIdWsf2PsRemoveEntityRequest object

Types and Values

struct LassoIdWsf2PsRemoveEntityRequest

```
struct LassoIdWsf2PsRemoveEntityRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  GList *TargetObjectID; /* of LassoNode */
};
```

11.82 LassoIdWsf2PsRemoveFromCollectionRequest

LassoIdWsf2PsRemoveFromCollectionRequest — <ps:RemoveFromCollectionRequest>

Functions

LassoIdWsf2PsRemoveFromCollectionRequest *	lasso_idwsf2_ps_remove_from_collection_request_new ()

Types and Values

struct	LassoIdWsf2PsRemoveFromCollectionRequest

Description

```
<xs:complexType name="RemoveFromCollectionRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID"/>
        <xs:element ref="ObjectID" maxOccurs="unbounded"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.81: Schema fragment for ps:RemoveFromCollectionRequest

Functions

lasso_idwsf2_ps_remove_from_collection_request_new ()

```
LassoIdWsf2PsRemoveFromCollectionRequest~*
lasso_idwsf2_ps_remove_from_collection_request_new
                                (void);
```

Creates a new LassoIdWsf2PsRemoveFromCollectionRequest object.

Returns

a newly created LassoIdWsf2PsRemoveFromCollectionRequest object

Types and Values

struct LassoIdWsf2PsRemoveFromCollectionRequest

```
struct LassoIdWsf2PsRemoveFromCollectionRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *TargetObjectID;
  GList *ObjectID; /* of LassoNode */
  /* XXX */ void *Subscription;
};
```

11.83 LassoIdWsf2PsRequestAbstract

LassoIdWsf2PsRequestAbstract — <ps:RequestAbstract>

Functions

LassoIdWsf2PsRequestAbstract *	lasso_idwsf2_ps_request_abstract_new ()

Types and Values

struct	LassoIdWsf2PsRequestAbstract

Description

```
<xs:complexType name="RequestAbstractType" abstract="true">
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.82: Schema fragment for ps:RequestAbstract

Functions

lasso_idwsf2_ps_request_abstract_new ()

```
LassoIdWsf2PsRequestAbstract~*
lasso_idwsf2_ps_request_abstract_new (void);
```

Creates a new LassoIdWsf2PsRequestAbstract object.

Returns

a newly created LassoIdWsf2PsRequestAbstract object

Types and Values

struct LassoIdWsf2PsRequestAbstract

```
struct LassoIdWsf2PsRequestAbstract {
  LassoNode parent;

  /* attributes */
  GHashTable *attributes;
};
```

11.84 LassoIdWsf2PsResolveIdentifierRequest

LassoIdWsf2PsResolveIdentifierRequest — <ps:ResolveIdentifierRequest>

Functions

LassoIdWsf2PsResolveIdentifierRequest *	lasso_idwsf2_ps_resolve_identifier_request_new ()

Types and Values

struct	LassoIdWsf2PsResolveIdentifierRequest

Description

```
<xs:complexType name="ResolveIdentifierRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="ResolveInput" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.83: Schema fragment for ps:ResolveIdentifierRequest

Functions

lasso_idwsf2_ps_resolve_identifier_request_new ()

```
LassoIdWsf2PsResolveIdentifierRequest~*
lasso_idwsf2_ps_resolve_identifier_request_new
                               (void);
```

Creates a new LassoIdWsf2PsResolveIdentifierRequest object.

Returns

a newly created LassoIdWsf2PsResolveIdentifierRequest object

Types and Values

struct LassoIdWsf2PsResolveIdentifierRequest

```
struct LassoIdWsf2PsResolveIdentifierRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  GList *ResolveInput; /* of LassoNode */
};
```

11.85 LassoIdWsf2PsResolveIdentifierResponse

LassoIdWsf2PsResolveIdentifierResponse — <ps:ResolveIdentifierResponse>

Functions

LassoIdWsf2PsResolveIdentifierResponse *	lasso_idwsf2_ps_resolve_identifier_response_new ()

Types and Values

struct	LassoIdWsf2PsResolveIdentifierResponse

Description

```
<xs:complexType name="ResolveIdentifierResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element ref="ResolveOutput" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.84: Schema fragment for ps:ResolveIdentifierResponse

Functions

lasso_idwsf2_ps_resolve_identifier_response_new ()

```
LassoIdWsf2PsResolveIdentifierResponse~*
lasso_idwsf2_ps_resolve_identifier_response_new
                                (void);
```

Creates a new LassoIdWsf2PsResolveIdentifierResponse object.

Returns

a newly created LassoIdWsf2PsResolveIdentifierResponse object

Types and Values

struct LassoIdWsf2PsResolveIdentifierResponse

```
struct LassoIdWsf2PsResolveIdentifierResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  GList *ResolveOutput; /* of LassoNode */
};
```

11.86 LassoIdWsf2PsResolveInput

LassoIdWsf2PsResolveInput — <ps:ResolveInput>

Functions

LassoIdWsf2PsResolveInput *	lasso_idwsf2_ps_resolve_input_new ()

Types and Values

struct	LassoIdWsf2PsResolveInput

Description

```
<xs:complexType name="ResolveInputType">
  <xs:complexContent>
    <xs:extension base="ims:MappingInputType">
      <xs:sequence>
        <xs:element ref="TargetObjectID" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.85: Schema fragment for ps:ResolveInput

Functions

lasso_idwsf2_ps_resolve_input_new ()

```
LassoIdWsf2PsResolveInput~*
lasso_idwsf2_ps_resolve_input_new (void);
```

Creates a new LassoIdWsf2PsResolveInput object.

Returns

a newly created LassoIdWsf2PsResolveInput object

Types and Values

struct LassoIdWsf2PsResolveInput

```
struct LassoIdWsf2PsResolveInput {
  LassoIdWsf2ImsMappingInput parent;

  /* elements */
  /* XXX */ void *TargetObjectID;
};
```

11.87 LassoIdWsf2PsResponseAbstract

LassoIdWsf2PsResponseAbstract — <ps:ResponseAbstract>

Functions

LassoIdWsf2PsResponseAbstract *	lasso_idwsf2_ps_response_abstract_new ()

Types and Values

struct	LassoIdWsf2PsResponseAbstract

Description

```
<xs:complexType name="ResponseAbstractType" abstract="true">
  <xs:sequence>
    <xs:element ref="lu:Status"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.86: Schema fragment for ps:ResponseAbstract

Functions

lasso_idwsf2_ps_response_abstract_new ()

```
LassoIdWsf2PsResponseAbstract~*
lasso_idwsf2_ps_response_abstract_new (void);
```

Creates a new LassoIdWsf2PsResponseAbstract object.

Returns

a newly created LassoIdWsf2PsResponseAbstract object

Types and Values

struct LassoIdWsf2PsResponseAbstract

```
struct LassoIdWsf2PsResponseAbstract {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  /* attributes */
  GHashTable *attributes;
};
```

11.88 LassoIdWsf2PsSetObjectInfoRequest

LassoIdWsf2PsSetObjectInfoRequest — <ps:SetObjectInfoRequest>

Functions

LassoIdWsf2PsSetObjectInfoRequest *	lasso_idwsf2_ps_set_object_info_request_new ()

Types and Values

struct	LassoIdWsf2PsSetObjectInfoRequest

Description

```
<xs:complexType name="SetObjectInfoRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="Object" maxOccurs="unbounded"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.87: Schema fragment for ps:SetObjectInfoRequest

Functions

lasso_idwsf2_ps_set_object_info_request_new ()

```
LassoIdWsf2PsSetObjectInfoRequest~*
lasso_idwsf2_ps_set_object_info_request_new
                               (void);
```

Creates a new LassoIdWsf2PsSetObjectInfoRequest object.

Returns

a newly created LassoIdWsf2PsSetObjectInfoRequest object

Types and Values

struct LassoIdWsf2PsSetObjectInfoRequest

```
struct LassoIdWsf2PsSetObjectInfoRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  GList *Object; /* of LassoNode */
  /* XXX */ void *Subscription;
};
```

11.89 LassoIdWsf2PsTestMembershipRequest

LassoIdWsf2PsTestMembershipRequest — <ps:TestMembershipRequest>

Functions

LassoIdWsf2PsTestMembershipRequest *	lasso_idwsf2_ps_test_membership_request_new ()

Types and Values

struct	LassoIdWsf2PsTestMembershipRequest

Description

```
<xs:complexType name="TestMembershipRequestType">
  <xs:complexContent>
    <xs:extension base="RequestAbstractType">
      <xs:sequence>
        <xs:element ref="TargetObjectID" minOccurs="0"/>
        <xs:element ref="sec:Token"/>
        <xs:element ref="Subscription" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.88: Schema fragment for ps:TestMembershipRequest

Functions

lasso_idwsf2_ps_test_membership_request_new ()

```
LassoIdWsf2PsTestMembershipRequest~*
lasso_idwsf2_ps_test_membership_request_new
                               (void);
```

Creates a new LassoIdWsf2PsTestMembershipRequest object.

Returns

a newly created LassoIdWsf2PsTestMembershipRequest object

Types and Values

struct LassoIdWsf2PsTestMembershipRequest

```
struct LassoIdWsf2PsTestMembershipRequest {
  LassoIdWsf2PsRequestAbstract parent;

  /* elements */
  /* XXX */ void *TargetObjectID;
  /* XXX */ void *Token;
  /* XXX */ void *Subscription;
};
```

11.90 LassoIdWsf2PsTestMembershipResponse

LassoIdWsf2PsTestMembershipResponse — <ps:TestMembershipResponse>

Functions

LassoIdWsf2PsTestMembershipResponse *	lasso_idwsf2_ps_test_membership_response_new ()

Types and Values

struct	LassoIdWsf2PsTestMembershipResponse

Description

```
<xs:complexType name="TestMembershipResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseAbstractType">
      <xs:sequence>
        <xs:element name="Result" type="ResultType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.89: Schema fragment for ps:TestMembershipResponse

Functions

lasso_idwsf2_ps_test_membership_response_new ()

```
LassoIdWsf2PsTestMembershipResponse~*
lasso_idwsf2_ps_test_membership_response_new
                               (void);
```

Creates a new LassoIdWsf2PsTestMembershipResponse object.

Returns

a newly created LassoIdWsf2PsTestMembershipResponse object

Types and Values

struct LassoIdWsf2PsTestMembershipResponse

```
struct LassoIdWsf2PsTestMembershipResponse {
  LassoIdWsf2PsResponseAbstract parent;

  /* elements */
  /* XXX */ void *Result;
};
```

11.91 LassoIdWsf2Sb2Consent

LassoIdWsf2Sb2Consent — <sb2:Consent>

Functions

LassoIdWsf2Sb2Consent *	lasso_idwsf2_sb2_consent_new ()

Types and Values

struct	LassoIdWsf2Sb2Consent

Description

```
<xs:complexType name="ConsentType">
  <xs:attribute name="uri" type="xs:anyURI" use="required"/>
  <xs:attribute name="timestamp" type="xs:dateTime" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.90: Schema fragment for sb2:Consent

Functions

lasso_idwsf2_sb2_consent_new ()

```
LassoIdWsf2Sb2Consent~*
lasso_idwsf2_sb2_consent_new (void);
```

Creates a new LassoIdWsf2Sb2Consent object.

Returns

a newly created LassoIdWsf2Sb2Consent object

Types and Values

struct LassoIdWsf2Sb2Consent

```
struct LassoIdWsf2Sb2Consent {
  LassoNode parent;

  /* attributes */
  char *uri;
  char *timestamp;
  GHashTable *attributes;
};
```

11.92 LassoIdWsf2Sb2CredentialsContext

LassoIdWsf2Sb2CredentialsContext — <sb2:CredentialsContext>

Functions

LassoIdWsf2Sb2CredentialsContext *	lasso_idwsf2_sb2_credentials_context_new ()

Types and Values

struct	LassoIdWsf2Sb2CredentialsContext

Description

```
<xs:complexType name="CredentialsContextType">
  <xs:sequence>
    <xs:element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
    <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded ↩
       "/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.91: Schema fragment for sb2:CredentialsContext

Functions

lasso_idwsf2_sb2_credentials_context_new ()

```
LassoIdWsf2Sb2CredentialsContext~*
lasso_idwsf2_sb2_credentials_context_new
                               (void);
```

Creates a new LassoIdWsf2Sb2CredentialsContext object.

Returns

a newly created LassoIdWsf2Sb2CredentialsContext object

Types and Values

struct LassoIdWsf2Sb2CredentialsContext

```
struct LassoIdWsf2Sb2CredentialsContext {
  LassoNode parent;

  /* elements */
  LassoSamlp2RequestedAuthnContext *RequestedAuthnContext;
  GList *SecurityMechID; /* of strings */
  /* attributes */
  GHashTable *attributes;
};
```

11.93 LassoIdWsf2Sb2EndpointUpdate

LassoIdWsf2Sb2EndpointUpdate — <sb2:EndpointUpdate>

Functions

LassoIdWsf2Sb2EndpointUpdate *	lasso_idwsf2_sb2_endpoint_update_new ()

Types and Values

struct	LassoIdWsf2Sb2EndpointUpdate

Description

```
<xs:complexType name="EndpointUpdateType">
  <xs:complexContent>
    <xs:extension base="wsa:EndpointReferenceType">
      <xs:attribute name="updateType" type="xs:anyURI" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.92: Schema fragment for sb2:EndpointUpdate

Functions

lasso_idwsf2_sb2_endpoint_update_new ()

```
LassoIdWsf2Sb2EndpointUpdate~*
lasso_idwsf2_sb2_endpoint_update_new (void);
```

Creates a new LassoIdWsf2Sb2EndpointUpdate object.

Returns

a newly created LassoIdWsf2Sb2EndpointUpdate object

Types and Values

struct LassoIdWsf2Sb2EndpointUpdate

```
struct LassoIdWsf2Sb2EndpointUpdate {
  LassoWsAddrEndpointReference parent;

  /* attributes */
  char *updateType;
};
```

11.94 LassoIdWsf2Sb2RedirectRequest

LassoIdWsf2Sb2RedirectRequest — <sb2:RedirectRequest>

Functions

LassoIdWsf2Sb2RedirectRequest *	lasso_idwsf2_sb2_redirect_request_new ()
LassoIdWsf2Sb2RedirectRequest *	lasso_idwsf2_sb2_redirect_request_new_full ()

Types and Values

struct	LassoIdWsf2Sb2RedirectRequest

Description

```
<xs:complexType name="RedirectRequestType">
  <xs:attribute name="redirectURL" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Figure 11.93: Schema fragment for sb2:RedirectRequest

Functions

lasso_idwsf2_sb2_redirect_request_new ()

```
LassoIdWsf2Sb2RedirectRequest~*
lasso_idwsf2_sb2_redirect_request_new (void);
```

Creates a new LassoIdWsf2Sb2RedirectRequest object.

Returns

a newly created LassoIdWsf2Sb2RedirectRequest object

lasso_idwsf2_sb2_redirect_request_new_full ()

```
LassoIdWsf2Sb2RedirectRequest~*
lasso_idwsf2_sb2_redirect_request_new_full
                               (const gchar *redirect_url);
```

Types and Values

struct LassoIdWsf2Sb2RedirectRequest

```
struct LassoIdWsf2Sb2RedirectRequest {
  LassoNode parent;

  /* attributes */
  char *redirectURL;
};
```

11.95 LassoIdWsf2Sb2Sender

LassoIdWsf2Sb2Sender — <sb2:Sender>

Functions

LassoIdWsf2Sb2Sender *	lasso_idwsf2_sb2_sender_new ()

Types and Values

struct	LassoIdWsf2Sb2Sender

Description

```
<xs:complexType name="SenderType">
  <xs:attribute name="providerID" type="xs:anyURI" use="required"/>
  <xs:attribute name="affiliationID" type="xs:anyURI" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.94: Schema fragment for sb2:Sender

Functions

lasso_idwsf2_sb2_sender_new ()

```
LassoIdWsf2Sb2Sender~*
lasso_idwsf2_sb2_sender_new (void);
```

Creates a new LassoIdWsf2Sb2Sender object.

Returns

a newly created LassoIdWsf2Sb2Sender object

Types and Values

struct LassoIdWsf2Sb2Sender

```
struct LassoIdWsf2Sb2Sender {
  LassoNode parent;

  /* attributes */
  char *providerID;
  char *affiliationID;
  GHashTable *attributes;
};
```

11.96 LassoIdWsf2Sb2TargetIdentity

LassoIdWsf2Sb2TargetIdentity — <sb2:TargetIdentity>

Functions

LassoIdWsf2Sb2TargetIdentity *	lasso_idwsf2_sb2_target_identity_new ()

Types and Values

struct	LassoIdWsf2Sb2TargetIdentity

Description

```
<xs:complexType name="TargetIdentityType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.95: Schema fragment for sb2:TargetIdentity

Functions

lasso_idwsf2_sb2_target_identity_new ()

```
LassoIdWsf2Sb2TargetIdentity~*
lasso_idwsf2_sb2_target_identity_new (void);
```

Creates a new LassoIdWsf2Sb2TargetIdentity object.

Returns

a newly created LassoIdWsf2Sb2TargetIdentity object

Types and Values

struct LassoIdWsf2Sb2TargetIdentity

```
struct LassoIdWsf2Sb2TargetIdentity {
  LassoNode parent;

  /* attributes */
  GHashTable *attributes;
  GList *any; /* of LassoNode */
};
```

11.97 LassoIdWsf2Sb2Timeout

LassoIdWsf2Sb2Timeout — <sb2:Timeout>

Functions

LassoIdWsf2Sb2Timeout *	lasso_idwsf2_sb2_timeout_new ()

Types and Values

struct	LassoIdWsf2Sb2Timeout

Description

```
<xs:complexType name="TimeoutType">
  <xs:attribute name="maxProcessingTime" type="xs:integer" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.96: Schema fragment for sb2:Timeout

Functions

lasso_idwsf2_sb2_timeout_new ()

```
LassoIdWsf2Sb2Timeout~*
lasso_idwsf2_sb2_timeout_new (void);
```

Creates a new LassoIdWsf2Sb2Timeout object.

Returns

a newly created LassoIdWsf2Sb2Timeout object

Types and Values

struct LassoIdWsf2Sb2Timeout

```
struct LassoIdWsf2Sb2Timeout {
  LassoNode parent;

  /* attributes */
  int maxProcessingTime;
  GHashTable *attributes;
};
```

11.98 LassoIdWsf2Sb2UsageDirective

LassoIdWsf2Sb2UsageDirective — <sb2:UsageDirective>

Functions

LassoIdWsf2Sb2UsageDirective *	lasso_idwsf2_sb2_usage_directive_new ()

Types and Values

struct	LassoIdWsf2Sb2UsageDirective

Description

```
<xs:complexType name="UsageDirectiveType">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax"
      maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ref" type="xs:IDREF" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

Figure 11.97: Schema fragment for sb2:UsageDirective

Functions

lasso_idwsf2_sb2_usage_directive_new ()

```
LassoIdWsf2Sb2UsageDirective~*
lasso_idwsf2_sb2_usage_directive_new (void);
```

Creates a new LassoIdWsf2Sb2UsageDirective object.

Returns

a newly created LassoIdWsf2Sb2UsageDirective object

Types and Values

struct LassoIdWsf2Sb2UsageDirective

```
struct LassoIdWsf2Sb2UsageDirective {
  LassoNode parent;

  /* attributes */
  char *ref;
  GHashTable *attributes;
};
```

11.99 LassoIdWsf2Sb2UserInteractionHeader

LassoIdWsf2Sb2UserInteractionHeader — <sb2:UserInteractionHeader>

Functions

LassoIdWsf2Sb2UserInteractionHeader *	lasso_idwsf2_sb2_user_interaction_header_new ()

Types and Values

struct	LassoIdWsf2Sb2UserInteractionHeader

Description

```
<xs:complexType name="UserInteractionHeaderType">
  <xs:sequence>
    <xs:element name="InteractionService" type="wsa:EndpointReferenceType"
            minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="interact" type="xs:string" use="optional"
          default="interactIfNeeded"/>
  <xs:attribute name="language" type="xs:NMTOKENS" use="optional"/>
  <xs:attribute name="redirect" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="maxInteractTime" type="xs:integer" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.98: Schema fragment for sb2:UserInteractionHeader

Functions

lasso_idwsf2_sb2_user_interaction_header_new ()

```
LassoIdWsf2Sb2UserInteractionHeader~*
lasso_idwsf2_sb2_user_interaction_header_new
                                (void);
```

Creates a new LassoIdWsf2Sb2UserInteractionHeader object.

Returns

a newly created LassoIdWsf2Sb2UserInteractionHeader object

Types and Values

struct LassoIdWsf2Sb2UserInteractionHeader

```
struct LassoIdWsf2Sb2UserInteractionHeader {
  LassoNode parent;

  /* elements */
  GList *InteractionService; /* of LassoNode */
  /* attributes */
  char *interact;
  char *language;
  gboolean redirect;
  int maxInteractTime;
  GHashTable *attributes;
};
```

11.100 LassoIdWsf2SbfFramework

LassoIdWsf2SbfFramework — <sbf:Framework>

Functions

LassoIdWsf2SbfFramework *	lasso_idwsf2_sbf_framework_new ()
LassoIdWsf2SbfFramework *	lasso_idwsf2_sbf_framework_new_full ()

Types and Values

struct	LassoIdWsf2SbfFramework

Description

```
<xs:complexType name="FrameworkType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.99: Schema fragment for sbf:Framework

Functions

lasso_idwsf2_sbf_framework_new ()

```
LassoIdWsf2SbfFramework~*
lasso_idwsf2_sbf_framework_new (void);
```

Creates a new LassoIdWsf2SbfFramework object.

Returns

a newly created LassoIdWsf2SbfFramework object

lasso_idwsf2_sbf_framework_new_full ()

```
LassoIdWsf2SbfFramework~*
lasso_idwsf2_sbf_framework_new_full (const char *version);
```

Types and Values

struct LassoIdWsf2SbfFramework

```
struct LassoIdWsf2SbfFramework {
  LassoNode parent;

  /* attributes */
  char *version;
  GHashTable *attributes;
};
```

11.101 LassoIdWsf2SecTokenPolicy

LassoIdWsf2SecTokenPolicy — <sec:TokenPolicy>

Functions

LassoIdWsf2SecTokenPolicy *	lasso_idwsf2_sec_token_policy_new ()

Types and Values

struct	LassoIdWsf2SecTokenPolicy

Description

```
<xs:complexType name="TokenPolicyType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="validUntil" type="xs:dateTime" use="optional"/>
  <xs:attribute name="issueTo" type="xs:anyURI" use="optional"/>
  <xs:attribute name="type" type="xs:anyURI" use="optional"/>
  <xs:attribute name="wantDSEPR" type="xs:boolean" use="optional" />
  <xs:anyAttribute namespace="##other" processContents="lax" />
</xs:complexType>
```

Figure 11.100: Schema fragment for sec:TokenPolicy

Functions

lasso_idwsf2_sec_token_policy_new ()

```
LassoIdWsf2SecTokenPolicy~*
lasso_idwsf2_sec_token_policy_new (void);
```

Creates a new LassoIdWsf2SecTokenPolicy object.

Returns

a newly created LassoIdWsf2SecTokenPolicy object

Types and Values

struct LassoIdWsf2SecTokenPolicy

```
struct LassoIdWsf2SecTokenPolicy {
  LassoNode parent;

  /* elements */
  LassoNode *any;
  /* attributes */
  char *validUntil;
  char *issueTo;
  char *type;
  gboolean wantDSEPR;
  GHashTable *attributes;
};
```

11.102 LassoIdWsf2SecToken

LassoIdWsf2SecToken — <sec:Token>

Functions

LassoIdWsf2SecToken *	lasso_idwsf2_sec_token_new ()

Types and Values

struct	LassoIdWsf2SecToken

Description

```
<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="optional" />
    <xs:attribute name="ref" type="xs:anyURI" use="optional" />
    <xs:attribute name="usage" type="xs:anyURI" use="optional" />
  </xs:complexType>
```

Figure 11.101: Schema fragment for sec:Token

Functions

lasso_idwsf2_sec_token_new ()

```
LassoIdWsf2SecToken~*
lasso_idwsf2_sec_token_new (void);
```

Creates a new LassoIdWsf2SecToken object.

Returns

a newly created LassoIdWsf2SecToken object

Types and Values

struct LassoIdWsf2SecToken

```
struct LassoIdWsf2SecToken {
  LassoNode parent;

  /* elements */
  LassoNode *any;
  /* attributes */
  char *id;
  char *ref;
  char *usage;
};
```

11.103 LassoIdWsf2SecTransitedProviderPath

LassoIdWsf2SecTransitedProviderPath — <sec:TransitedProviderPath>

Functions

LassoIdWsf2SecTransitedProviderPath	*lasso_idwsf2_sec_transited_provider_path_new ()

Types and Values

struct	LassoIdWsf2SecTransitedProviderPath

Description

```
<xs:complexType name="TransitedProviderPathType">
  <xs:sequence>
    <xs:element ref="sec:TransitedProvider" minOccurs="1"
      maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
```

Figure 11.102: Schema fragment for sec:TransitedProviderPath

Functions

lasso_idwsf2_sec_transited_provider_path_new ()

```
LassoIdWsf2SecTransitedProviderPath~*
lasso_idwsf2_sec_transited_provider_path_new
                                (void);
```

Creates a new LassoIdWsf2SecTransitedProviderPath object.

Returns

a newly created LassoIdWsf2SecTransitedProviderPath object

Types and Values

struct LassoIdWsf2SecTransitedProviderPath

```
struct LassoIdWsf2SecTransitedProviderPath {
  LassoNode parent;

  /* elements */
  GList *TransitedProvider; /* of LassoNode */
};
```

11.104 LassoIdWsf2SubsNotification

LassoIdWsf2SubsNotification — <subs:Notification>

Functions

LassoIdWsf2SubsNotification *	lasso_idwsf2_subs_notification_new ()

Types and Values

struct	LassoIdWsf2SubsNotification

Description

```
<xs:complexType name="NotificationType">
  <xs:sequence>
    <xs:element ref="lu:TestResult" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" use="optional" type="xs:ID"/>
  <xs:attribute name="subscriptionID" use="required" type="lu:IDType"/>
  <xs:attribute name="expires" use="optional" type="xs:dateTime"/>
  <xs:attribute name="endReason" use="optional" type="xs:anyURI"/>
</xs:complexType>
```

Figure 11.103: Schema fragment for subs:Notification

Functions

lasso_idwsf2_subs_notification_new ()

```
LassoIdWsf2SubsNotification~*
lasso_idwsf2_subs_notification_new (void);
```

Creates a new LassoIdWsf2SubsNotification object.

Returns

a newly created LassoIdWsf2SubsNotification object

Types and Values

struct LassoIdWsf2SubsNotification

```
struct LassoIdWsf2SubsNotification {
  LassoNode parent;

  /* elements */
  GList *TestResult; /* of LassoNode */
  /* attributes */
  char *id;
  char *subscriptionID;
  char *expires;
  char *endReason;
};
```

11.105 LassoIdWsf2SubsNotifyResponse

LassoIdWsf2SubsNotifyResponse — <subs:NotifyResponse>

Functions

LassoIdWsf2SubsNotifyResponse *	lasso_idwsf2_subs_notify_response_new ()

Types and Values

struct	LassoIdWsf2SubsNotifyResponse

Description

```
<xs:complexType name="NotifyResponseType">
  <xs:complexContent>
    <xs:extension base="lu:ResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.104: Schema fragment for subs:NotifyResponse

Functions

lasso_idwsf2_subs_notify_response_new ()

```
LassoIdWsf2SubsNotifyResponse~*
lasso_idwsf2_subs_notify_response_new (void);
```

Creates a new LassoIdWsf2SubsNotifyResponse object.

Returns

a newly created LassoIdWsf2SubsNotifyResponse object

Types and Values

struct LassoIdWsf2SubsNotifyResponse

```
struct LassoIdWsf2SubsNotifyResponse {
  LassoIdWsf2UtilResponse parent;
};
```

11.106 LassoIdWsf2SubsRefAppData

LassoIdWsf2SubsRefAppData — <subsref:AppData>

Functions

LassoIdWsf2SubsRefAppData *	lasso_idwsf2_subsref_app_data_new ()

Types and Values

struct	LassoIdWsf2SubsRefAppData

Description

Functions

lasso_idwsf2_subsref_app_data_new ()

```
LassoIdWsf2SubsRefAppData~*
lasso_idwsf2_subsref_app_data_new (void);
```

Creates a new LassoIdWsf2SubsRefAppData object.

Returns

a newly created LassoIdWsf2SubsRefAppData object

Types and Values

struct LassoIdWsf2SubsRefAppData

```
struct LassoIdWsf2SubsRefAppData {
  LassoNode parent;

  /* elements */
  GList *any; /* of xmlNode* */
  GList *Subscription; /* of LassoNode */
};
```

11.107 LassoIdWsf2SubsRefCreateItem

LassoIdWsf2SubsRefCreateItem — <subsref:CreateItem>

Functions

LassoIdWsf2SubsRefCreateItem *	lasso_idwsf2_subsref_create_item_new ()

Types and Values

struct	LassoIdWsf2SubsRefCreateItem

Description

```
<xs:complexType name="CreateItemType">
  <xs:sequence>
    <xs:element ref="subsref:NewData" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="dst:CreateItemAttributeGroup"/>
</xs:complexType>
```

Figure 11.105: Schema fragment for subsref:CreateItem

Functions

lasso_idwsf2_subsref_create_item_new ()

```
LassoIdWsf2SubsRefCreateItem~*
lasso_idwsf2_subsref_create_item_new (void);
```

Creates a new LassoIdWsf2SubsRefCreateItem object.

Returns

a newly created LassoIdWsf2SubsRefCreateItem object

Types and Values

struct LassoIdWsf2SubsRefCreateItem

```
struct LassoIdWsf2SubsRefCreateItem {
  LassoNode parent;

  /* elements */
  /* XXX */ void *NewData;
  /* attributes */
  char *objectType;
  char *id;
  char *itemID;
};
```

11.108 LassoIdWsf2SubsRefCreateResponse

LassoIdWsf2SubsRefCreateResponse — <subsref:CreateResponse>

Functions

LassoIdWsf2SubsRefCreateResponse * lasso_idwsf2_subsref_create_response_new ()

Types and Values

struct	LassoIdWsf2SubsRefCreateResponse

Description

```
<xs:complexType name="CreateResponseType">
  <xs:complexContent>
    <xs:extension base="subsref:DataResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.106: Schema fragment for subsref:CreateResponse

Functions

lasso_idwsf2_subsref_create_response_new ()

```
LassoIdWsf2SubsRefCreateResponse~*
lasso_idwsf2_subsref_create_response_new
                               (void);
```

Creates a new LassoIdWsf2SubsRefCreateResponse object.

Returns

a newly created LassoIdWsf2SubsRefCreateResponse object

Types and Values

struct LassoIdWsf2SubsRefCreateResponse

```
struct LassoIdWsf2SubsRefCreateResponse {
  LassoIdWsf2SubsRefDataResponse parent;
};
```

11.109 LassoIdWsf2SubsRefCreate

LassoIdWsf2SubsRefCreate — <subsref:Create>

Functions

LassoIdWsf2SubsRefCreate *	lasso_idwsf2_subsref_create_new ()

Types and Values

struct	LassoIdWsf2SubsRefCreate

Description

```
<xs:complexType name="CreateType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="subsref:Subscription" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="subsref:CreateItem" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="subsref:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.107: Schema fragment for subsref:Create

Functions

lasso_idwsf2_subsref_create_new ()

```
LassoIdWsf2SubsRefCreate~*
lasso_idwsf2_subsref_create_new (void);
```

Creates a new LassoIdWsf2SubsRefCreate object.

Returns

a newly created LassoIdWsf2SubsRefCreate object

Types and Values

struct LassoIdWsf2SubsRefCreate

```
struct LassoIdWsf2SubsRefCreate {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *Subscription; /* of LassoNode */
  GList *CreateItem; /* of LassoNode */
  GList *ResultQuery; /* of LassoNode */
};
```

11.110 LassoIdWsf2SubsRefDataResponse

LassoIdWsf2SubsRefDataResponse — <subsref:DataResponse>

Functions

LassoIdWsf2SubsRefDataResponse *	lasso_idwsf2_subsref_data_response_new ()

Types and Values

struct	LassoIdWsf2SubsRefDataResponse

Description

```
<xs:complexType name="DataResponseType">
  <xs:complexContent>
    <xs:extension base="dst:DataResponseBaseType">
      <xs:sequence>
        <xs:element ref="subsref:ItemData" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.108: Schema fragment for subsref:DataResponse

Functions

lasso_idwsf2_subsref_data_response_new ()

```
LassoIdWsf2SubsRefDataResponse~*
lasso_idwsf2_subsref_data_response_new
                               (void);
```

Creates a new LassoIdWsf2SubsRefDataResponse object.

Returns

a newly created LassoIdWsf2SubsRefDataResponse object

Types and Values

struct LassoIdWsf2SubsRefDataResponse

```
struct LassoIdWsf2SubsRefDataResponse {
  LassoIdWsf2DstDataResponseBase parent;

  /* elements */
  GList *ItemData; /* of LassoNode */
};
```

11.111 LassoIdWsf2SubsRefData

LassoIdWsf2SubsRefData — <subsref:Data>

Functions

LassoIdWsf2SubsRefData *	lasso_idwsf2_subsref_data_new ()

Types and Values

struct	LassoIdWsf2SubsRefData

Description

```
<xs:complexType name="DataType">
  <xs:complexContent>
    <xs:extension base="subsref:ItemDataType">
      <xs:attributeGroup ref="dst:PaginationResponseAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.109: Schema fragment for subsref:Data

Functions

lasso_idwsf2_subsref_data_new ()

```
LassoIdWsf2SubsRefData~*
lasso_idwsf2_subsref_data_new (void);
```

Creates a new LassoIdWsf2SubsRefData object.

Returns

a newly created LassoIdWsf2SubsRefData object

Types and Values

struct LassoIdWsf2SubsRefData

```
struct LassoIdWsf2SubsRefData {
  LassoIdWsf2SubsRefItemData parent;

  /* attributes */
  int remaining;
  int nextOffset;
  char *setID;
};
```

11.112 LassoIdWsf2SubsRefDeleteItem

LassoIdWsf2SubsRefDeleteItem — <subsref:DeleteItem>

Functions

LassoIdWsf2SubsRefDeleteItem *	lasso_idwsf2_subsref_delete_item_new ()

Types and Values

struct	LassoIdWsf2SubsRefDeleteItem

Description

```
<xs:complexType name="DeleteItemType">
  <xs:complexContent>
    <xs:extension base="dst:DeleteItemBaseType">
      <xs:sequence>
        <xs:element ref="subsref:Select" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.110: Schema fragment for subsref:DeleteItem

Functions

lasso_idwsf2_subsref_delete_item_new ()

```
LassoIdWsf2SubsRefDeleteItem~*
lasso_idwsf2_subsref_delete_item_new (void);
```

Creates a new LassoIdWsf2SubsRefDeleteItem object.

Returns

a newly created LassoIdWsf2SubsRefDeleteItem object

Types and Values

struct LassoIdWsf2SubsRefDeleteItem

```
struct LassoIdWsf2SubsRefDeleteItem {
  LassoIdWsf2DstDeleteItemBase parent;

  /* elements */
  /* XXX */ void *Select;
};
```

11.113 LassoIdWsf2SubsRefDeleteResponse

LassoIdWsf2SubsRefDeleteResponse — <subsref:DeleteResponse>

Functions

LassoIdWsf2SubsRefDeleteResponse *	lasso_idwsf2_subsref_delete_response_new ()

Types and Values

struct	LassoIdWsf2SubsRefDeleteResponse

Description

```
<xs:complexType name="DeleteResponseType">
  <xs:complexContent>
    <xs:extension base="lu:ResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.111: Schema fragment for subsref:DeleteResponse

Functions

lasso_idwsf2_subsref_delete_response_new ()

```
LassoIdWsf2SubsRefDeleteResponse~*
lasso_idwsf2_subsref_delete_response_new
                               (void);
```

Creates a new LassoIdWsf2SubsRefDeleteResponse object.

Returns

a newly created LassoIdWsf2SubsRefDeleteResponse object

Types and Values

struct LassoIdWsf2SubsRefDeleteResponse

```
struct LassoIdWsf2SubsRefDeleteResponse {
  LassoIdWsf2UtilResponse parent;
};
```

11.114 LassoIdWsf2SubsRefDelete

LassoIdWsf2SubsRefDelete — <subsref:Delete>

Functions

LassoIdWsf2SubsRefDelete *	lasso_idwsf2_subsref_delete_new ()

Types and Values

struct	LassoIdWsf2SubsRefDelete

Description

```
<xs:complexType name="DeleteType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="subsref:DeleteItem" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.112: Schema fragment for subsref:Delete

Functions

lasso_idwsf2_subsref_delete_new ()

```
LassoIdWsf2SubsRefDelete~*
lasso_idwsf2_subsref_delete_new (void);
```

Creates a new LassoIdWsf2SubsRefDelete object.

Returns

a newly created LassoIdWsf2SubsRefDelete object

Types and Values

struct LassoIdWsf2SubsRefDelete

```
struct LassoIdWsf2SubsRefDelete {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *DeleteItem; /* of LassoNode */
};
```

11.115 LassoIdWsf2SubsRefItemData

LassoIdWsf2SubsRefItemData — <subsref:ItemData>

Functions

LassoIdWsf2SubsRefItemData *	lasso_idwsf2_subsref_item_data_new ()

Types and Values

struct	LassoIdWsf2SubsRefItemData

Description

```
<xs:complexType name="ItemDataType">
  <xs:complexContent>
    <xs:extension base="subsref:AppDataType">
      <xs:attributeGroup ref="dst:ItemDataAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.113: Schema fragment for subsref:ItemData

Functions

lasso_idwsf2_subsref_item_data_new ()

```
LassoIdWsf2SubsRefItemData~*
lasso_idwsf2_subsref_item_data_new (void);
```

Creates a new LassoIdWsf2SubsRefItemData object.

Returns

a newly created LassoIdWsf2SubsRefItemData object

Types and Values

struct LassoIdWsf2SubsRefItemData

```
struct LassoIdWsf2SubsRefItemData {
  LassoIdWsf2SubsRefAppData parent;

  /* attributes */
  char *itemIDRef;
  char *notSorted;
  char *changeFormat;
};
```

11.116 LassoIdWsf2SubsRefItem

LassoIdWsf2SubsRefItem — <subs:RefItem>

Functions

LassoIdWsf2SubsRefItem *	lasso_idwsf2_subs_ref_item_new ()

Types and Values

struct	LassoIdWsf2SubsRefItem

Description

```
<xs:complexType name="RefItemType">
  <xs:attribute name="subscriptionID" use="optional" type="lu:IDType"/>
  <xs:attribute ref="lu:itemIDRef" use="required"/>
</xs:complexType>
```

Figure 11.114: Schema fragment for subs:RefItem

Functions

lasso_idwsf2_subs_ref_item_new ()

```
LassoIdWsf2SubsRefItem~*
lasso_idwsf2_subs_ref_item_new (void);
```

Creates a new LassoIdWsf2SubsRefItem object.

Returns

a newly created LassoIdWsf2SubsRefItem object

Types and Values

struct LassoIdWsf2SubsRefItem

```
struct LassoIdWsf2SubsRefItem {
  LassoNode parent;

  /* attributes */
  char *subscriptionID;
  char *itemIDRef;
};
```

11.117 LassoIdWsf2SubsRefModifyItem

LassoIdWsf2SubsRefModifyItem — <subsref:ModifyItem>

Functions

LassoIdWsf2SubsRefModifyItem *	lasso_idwsf2_subsref_modify_item_new ()

Types and Values

struct	LassoIdWsf2SubsRefModifyItem

Description

```
<xs:complexType name="ModifyItemType">
  <xs:sequence>
    <xs:element ref="subsref:Select" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="subsref:NewData" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="dst:ModifyItemAttributeGroup"/>
</xs:complexType>
```

Figure 11.115: Schema fragment for subsref:ModifyItem

Functions

lasso_idwsf2_subsref_modify_item_new ()

```
LassoIdWsf2SubsRefModifyItem~*
lasso_idwsf2_subsref_modify_item_new (void);
```

Creates a new LassoIdWsf2SubsRefModifyItem object.

Returns

a newly created LassoIdWsf2SubsRefModifyItem object

Types and Values

struct LassoIdWsf2SubsRefModifyItem

```
struct LassoIdWsf2SubsRefModifyItem {
  LassoNode parent;

  /* elements */
  /* XXX */ void *Select;
  /* XXX */ void *NewData;
  /* attributes */
  char *notChangedSince;
  gboolean overrideAllowed;
  char *id;
  char *itemID;
};
```

11.118 LassoIdWsf2SubsRefModifyResponse

LassoIdWsf2SubsRefModifyResponse — <subsref:ModifyResponse>

Functions

LassoIdWsf2SubsRefModifyResponse	*lasso_idwsf2_subsref_modify_response_new ()

Types and Values

struct	LassoIdWsf2SubsRefModifyResponse

Description

```
<xs:complexType name="ModifyResponseType">
  <xs:complexContent>
    <xs:extension base="subsref:DataResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.116: Schema fragment for subsref:ModifyResponse

Functions

lasso_idwsf2_subsref_modify_response_new ()

```
LassoIdWsf2SubsRefModifyResponse~*
lasso_idwsf2_subsref_modify_response_new
                                (void);
```

Creates a new LassoIdWsf2SubsRefModifyResponse object.

Returns

a newly created LassoIdWsf2SubsRefModifyResponse object

Types and Values

struct LassoIdWsf2SubsRefModifyResponse

```
struct LassoIdWsf2SubsRefModifyResponse {
  LassoIdWsf2SubsRefDataResponse parent;
};
```

11.119 LassoIdWsf2SubsRefModify

LassoIdWsf2SubsRefModify — <subsref:Modify>

Functions

LassoIdWsf2SubsRefModify *	lasso_idwsf2_subsref_modify_new ()

Types and Values

struct	LassoIdWsf2SubsRefModify

Description

```
<xs:complexType name="ModifyType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="subsref:Subscription" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="subsref:ModifyItem" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="subsref:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.117: Schema fragment for subsref:Modify

Functions

lasso_idwsf2_subsref_modify_new ()

```
LassoIdWsf2SubsRefModify~*
lasso_idwsf2_subsref_modify_new (void);
```

Creates a new LassoIdWsf2SubsRefModify object.

Returns

a newly created LassoIdWsf2SubsRefModify object

Types and Values

struct LassoIdWsf2SubsRefModify

```
struct LassoIdWsf2SubsRefModify {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *Subscription; /* of LassoNode */
  GList *ModifyItem; /* of LassoNode */
  GList *ResultQuery; /* of LassoNode */
};
```

11.120 LassoIdWsf2SubsRefNotification

LassoIdWsf2SubsRefNotification — <subsref:Notification>

Functions

LassoIdWsf2SubsRefNotification *	lasso_idwsf2_subsref_notification_new ()

Types and Values

struct	LassoIdWsf2SubsRefNotification

Description

```
<xs:complexType name="NotificationType">
  <xs:complexContent>
    <xs:extension base="subs:NotificationType">
      <xs:sequence>
        <xs:element ref="subsref:ItemData" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.118: Schema fragment for subsref:Notification

Functions

lasso_idwsf2_subsref_notification_new ()

```
LassoIdWsf2SubsRefNotification~*
lasso_idwsf2_subsref_notification_new (void);
```

Creates a new LassoIdWsf2SubsRefNotification object.

Returns

a newly created LassoIdWsf2SubsRefNotification object

Types and Values

struct LassoIdWsf2SubsRefNotification

```
struct LassoIdWsf2SubsRefNotification {
  LassoIdWsf2SubsNotification parent;

  /* elements */
  GList *ItemData; /* of LassoNode */
};
```

11.121 LassoIdWsf2SubsRefNotifyResponse

LassoIdWsf2SubsRefNotifyResponse — <subsref:NotifyResponse>

Functions

LassoIdWsf2SubsRefNotifyResponse *	lasso_idwsf2_subsref_notify_response_new ()

Types and Values

struct	LassoIdWsf2SubsRefNotifyResponse

Description

```
<xs:complexType name="NotifyResponseType">
  <xs:complexContent>
    <xs:extension base="subs:NotifyResponseType"/>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.119: Schema fragment for subsref:NotifyResponse

Functions

lasso_idwsf2_subsref_notify_response_new ()

```
LassoIdWsf2SubsRefNotifyResponse~*
lasso_idwsf2_subsref_notify_response_new
                               (void);
```

Creates a new LassoIdWsf2SubsRefNotifyResponse object.

Returns

a newly created LassoIdWsf2SubsRefNotifyResponse object

Types and Values

struct LassoIdWsf2SubsRefNotifyResponse

```
struct LassoIdWsf2SubsRefNotifyResponse {
  LassoIdWsf2SubsNotifyResponse parent;
};
```

11.122 LassoIdWsf2SubsRefNotify

LassoIdWsf2SubsRefNotify — <subsref:Notify>

Functions

LassoIdWsf2SubsRefNotify *	lasso_idwsf2_subsref_notify_new ()

Types and Values

struct	LassoIdWsf2SubsRefNotify

Description

```
<xs:complexType name="NotifyType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="subsref:Notification" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="subs:NotifyAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.120: Schema fragment for subsref:Notify

Functions

lasso_idwsf2_subsref_notify_new ()

```
LassoIdWsf2SubsRefNotify~*
lasso_idwsf2_subsref_notify_new (void);
```

Creates a new LassoIdWsf2SubsRefNotify object.

Returns

a newly created LassoIdWsf2SubsRefNotify object

Types and Values

struct LassoIdWsf2SubsRefNotify

```
struct LassoIdWsf2SubsRefNotify {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *Notification; /* of LassoNode */
  /* attributes */
  char *timeStamp;
};
```

11.123 LassoIdWsf2SubsRefQueryItem

LassoIdWsf2SubsRefQueryItem — <subsref:QueryItem>

Functions

LassoIdWsf2SubsRefQueryItem *	lasso_idwsf2_subsref_query_item_new ()

Types and Values

struct	LassoIdWsf2SubsRefQueryItem

Description

```
<xs:complexType name="QueryItemType">
  <xs:complexContent>
    <xs:extension base="subsref:ResultQueryType">
      <xs:attributeGroup ref="dst:PaginationAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.121: Schema fragment for subsref:QueryItem

Functions

lasso_idwsf2_subsref_query_item_new ()

```
LassoIdWsf2SubsRefQueryItem~*
lasso_idwsf2_subsref_query_item_new (void);
```

Creates a new LassoIdWsf2SubsRefQueryItem object.

Returns

a newly created LassoIdWsf2SubsRefQueryItem object

Types and Values

struct LassoIdWsf2SubsRefQueryItem

```
struct LassoIdWsf2SubsRefQueryItem {
  LassoIdWsf2SubsRefResultQuery parent;

  /* attributes */
  int count;
  int offset;
  char *setID;
  char *setReq;
};
```

11.124 LassoIdWsf2SubsRefQueryResponse

LassoIdWsf2SubsRefQueryResponse — <subsref:QueryResponse>

Functions

LassoIdWsf2SubsRefQueryResponse *	lasso_idwsf2_subsref_query_response_new ()

Types and Values

struct	LassoIdWsf2SubsRefQueryResponse

Description

```
<xs:complexType name="QueryResponseType">
  <xs:complexContent>
    <xs:extension base="dst:DataResponseBaseType">
      <xs:sequence>
        <xs:element ref="lu:TestResult" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="subsref:Data" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.122: Schema fragment for subsref:QueryResponse

Functions

lasso_idwsf2_subsref_query_response_new ()

```
LassoIdWsf2SubsRefQueryResponse~*
lasso_idwsf2_subsref_query_response_new
                               (void);
```

Creates a new LassoIdWsf2SubsRefQueryResponse object.

Returns

a newly created LassoIdWsf2SubsRefQueryResponse object

Types and Values

struct LassoIdWsf2SubsRefQueryResponse

```
struct LassoIdWsf2SubsRefQueryResponse {
  LassoIdWsf2DstDataResponseBase parent;
  /* elements */
  GList *TestResult; /* of LassoNode */
  GList *Data; /* of LassoNode */
};
```

11.125 LassoIdWsf2SubsRefQuery

LassoIdWsf2SubsRefQuery — <subsref:Query>

Functions

LassoIdWsf2SubsRefQuery *	lasso_idwsf2_subsref_query_new ()

Types and Values

struct	LassoIdWsf2SubsRefQuery

Description

```
<xs:complexType name="QueryType">
  <xs:complexContent>
    <xs:extension base="dst:RequestType">
      <xs:sequence>
        <xs:element ref="subsref:TestItem" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="subsref:QueryItem" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="subsref:Subscription" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.123: Schema fragment for subsref:Query

Functions

lasso_idwsf2_subsref_query_new ()

```
LassoIdWsf2SubsRefQuery~*
lasso_idwsf2_subsref_query_new (void);
```

Creates a new LassoIdWsf2SubsRefQuery object.

Returns

a newly created LassoIdWsf2SubsRefQuery object

Types and Values

struct LassoIdWsf2SubsRefQuery

```
struct LassoIdWsf2SubsRefQuery {
  LassoIdWsf2DstRequest parent;

  /* elements */
  GList *TestItem; /* of LassoNode */
  GList *QueryItem; /* of LassoNode */
  GList *Subscription; /* of LassoNode */
};
```

11.126 LassoIdWsf2SubsRefResultQuery

LassoIdWsf2SubsRefResultQuery — <subsref:ResultQuery>

Functions

LassoIdWsf2SubsRefResultQuery *	lasso_idwsf2_subsref_result_query_new ()

Types and Values

struct	LassoIdWsf2SubsRefResultQuery

Description

```
<xs:complexType name="ResultQueryType">
  <xs:complexContent>
    <xs:extension base="dst:ResultQueryBaseType">
      <xs:sequence>
        <xs:element ref="subsref:Select" minOccurs="0" maxOccurs="1"/>
        <xs:element name="Sort" minOccurs="0" maxOccurs="1" type="subsref:SortType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.124: Schema fragment for subsref:ResultQuery

Functions

lasso_idwsf2_subsref_result_query_new ()

```
LassoIdWsf2SubsRefResultQuery~*
lasso_idwsf2_subsref_result_query_new (void);
```

Creates a new LassoIdWsf2SubsRefResultQuery object.

Returns

a newly created LassoIdWsf2SubsRefResultQuery object

Types and Values

struct LassoIdWsf2SubsRefResultQuery

```
struct LassoIdWsf2SubsRefResultQuery {
  LassoIdWsf2DstResultQueryBase parent;

  /* elements */
  /* XXX */ void *Select;
  /* XXX */ void *Sort;
};
```

11.127 LassoIdWsf2SubsRefSubscription

LassoIdWsf2SubsRefSubscription — <subsref:Subscription>

Functions

LassoIdWsf2SubsRefSubscription *	lasso_idwsf2_subsref_subscription_new ()

Types and Values

struct	LassoIdWsf2SubsRefSubscription

Description

```
<xs:complexType name="SubscriptionType">
  <xs:complexContent>
    <xs:extension base="subs:SubscriptionType">
      <xs:sequence>
        <xs:element ref="subsref:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Aggregation" minOccurs="0" maxOccurs="1"
                type="subsref:AggregationType"/>
        <xs:element name="Trigger" minOccurs="0" maxOccurs="1" type="subsref:TriggerType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.125: Schema fragment for subsref:Subscription

Functions

lasso_idwsf2_subsref_subscription_new ()

```
LassoIdWsf2SubsRefSubscription~*
lasso_idwsf2_subsref_subscription_new (void);
```

Creates a new LassoIdWsf2SubsRefSubscription object.

Returns

a newly created LassoIdWsf2SubsRefSubscription object

Types and Values

struct LassoIdWsf2SubsRefSubscription

```
struct LassoIdWsf2SubsRefSubscription {
  LassoIdWsf2SubsSubscription parent;

  /* elements */
  GList *ResultQuery; /* of LassoNode */
  /* XXX */ void *Aggregation;
  /* XXX */ void *Trigger;
};
```

11.128 LassoIdWsf2SubsRefTestItem

LassoIdWsf2SubsRefTestItem — <subsref:TestItem>

Functions

LassoIdWsf2SubsRefTestItem *	lasso_idwsf2_subsref_test_item_new ()

Types and Values

struct	LassoIdWsf2SubsRefTestItem

Description

```
<xs:complexType name="TestItemType">
  <xs:complexContent>
    <xs:extension base="dst:TestItemBaseType">
      <xs:sequence>
        <xs:element name="TestOp" minOccurs="0" maxOccurs="1" type="subsref:TestOpType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 11.126: Schema fragment for subsref:TestItem

Functions

lasso_idwsf2_subsref_test_item_new ()

```
LassoIdWsf2SubsRefTestItem~*
lasso_idwsf2_subsref_test_item_new (void);
```

Creates a new LassoIdWsf2SubsRefTestItem object.

Returns

a newly created LassoIdWsf2SubsRefTestItem object

Types and Values

struct LassoIdWsf2SubsRefTestItem

```
struct LassoIdWsf2SubsRefTestItem {
  LassoIdWsf2DstTestItemBase parent;

  /* elements */
  /* XXX */ void *TestOp;
};
```

11.129 LassoIdWsf2SubsSubscription

LassoIdWsf2SubsSubscription — <subs:Subscription>

Functions

LassoIdWsf2SubsSubscription *	lasso_idwsf2_subs_subscription_new ()

Types and Values

struct	LassoIdWsf2SubsSubscription

Description

```
<xs:complexType name="SubscriptionType">
  <xs:sequence>
    <xs:element ref="subs:RefItem" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="lu:Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="subscriptionID" use="required" type="lu:IDType"/>
  <xs:attribute name="notifyToRef" use="required" type="xs:anyURI"/>
  <xs:attribute name="adminNotifyToRef" use="optional" type="xs:anyURI"/>
  <xs:attribute name="starts" use="optional" type="xs:dateTime"/>
  <xs:attribute name="expires" use="optional" type="xs:dateTime"/>
  <xs:attribute name="id" use="optional" type="xs:ID"/>
  <xs:attribute name="includeData" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Yes"/>
        <xs:enumeration value="No"/>
        <xs:enumeration value="YesWithCommonAttributes"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Figure 11.127: Schema fragment for subs:Subscription

Functions

lasso_idwsf2_subs_subscription_new ()

```
LassoIdWsf2SubsSubscription~*
lasso_idwsf2_subs_subscription_new (void);
```

Creates a new LassoIdWsf2SubsSubscription object.

Returns

a newly created LassoIdWsf2SubsSubscription object

Types and Values

struct LassoIdWsf2SubsSubscription

```
struct LassoIdWsf2SubsSubscription {
  LassoNode parent;

  /* elements */
  GList *RefItem; /* of LassoNode */
  GList *Extension; /* of LassoIdWsf2Utilextension */
  /* attributes */
  char *subscriptionID;
  char *notifyToRef;
  char *adminNotifyToRef;
  char *starts;
  char *expires;
  char *id;
  char *includeData;
};
```

11.130 LassoIdWsf2UtilEmpty

LassoIdWsf2UtilEmpty — <util:Empty>

Functions

LassoIdWsf2UtilEmpty *	lasso_idwsf2_util_empty_new ()

Types and Values

struct	LassoIdWsf2UtilEmpty

Description

```

<xs:complexType name="EmptyType">
  <xs:annotation>
    <xs:documentation> This type may be used to create an empty element </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:restriction base="xs:anyType"/>
    </xs:complexContent>
  </xs:complexType>
```

Figure 11.128: Schema fragment for util:Empty

Functions

lasso_idwsf2_util_empty_new ()

```
LassoIdWsf2UtilEmpty~*
lasso_idwsf2_util_empty_new (void);
```

Creates a new LassoIdWsf2UtilEmpty object.

Returns

a newly created LassoIdWsf2UtilEmpty object

Types and Values

struct LassoIdWsf2UtilEmpty

```
struct LassoIdWsf2UtilEmpty {
  LassoNode parent;
};
```

11.131 LassoIdWsf2Utilextension

LassoIdWsf2Utilextension — <util:extension>

Functions

LassoIdWsf2Utilextension *	lasso_idwsf2_util_extension_new ()

Types and Values

struct	LassoIdWsf2Utilextension

Description

```
<xs:complexType name="extensionType">
  <xs:annotation>
    <xs:documentation>
      A type for arbitrary content extensions from other namespaces
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Figure 11.129: Schema fragment for util:extension

Functions

lasso_idwsf2_util_extension_new ()

```
LassoIdWsf2Utilextension~*
lasso_idwsf2_util_extension_new (void);
```

Creates a new LassoIdWsf2Utilextension object.

Returns

a newly created LassoIdWsf2Utilextension object

Types and Values

struct LassoIdWsf2Utilextension

```
struct LassoIdWsf2Utilextension {
  LassoNode parent;
};
```

11.132 LassoIdWsf2UtilResponse

LassoIdWsf2UtilResponse — <util:Response>

Functions

LassoIdWsf2UtilResponse *	lasso_idwsf2_util_response_new ()
void	lasso_idwsf2_util_response_set_status ()
void	lasso_idwsf2_util_response_set_status2 ()

Types and Values

struct	LassoIdWsf2UtilResponse

Description

```
<xs:complexType name="ResponseType">
  <xs:sequence>
    <xs:element ref="Status"       minOccurs="1" maxOccurs="1"/>
    <xs:element ref="Extension"    minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="itemIDRef" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Figure 11.130: Schema fragment for util:Response

Functions

lasso_idwsf2_util_response_new ()

```
LassoIdWsf2UtilResponse~*
lasso_idwsf2_util_response_new (void);
```

Creates a new LassoIdWsf2UtilResponse object.

Returns

a newly created LassoIdWsf2UtilResponse object

lasso_idwsf2_util_response_set_status ()

```
void
lasso_idwsf2_util_response_set_status (LassoIdWsf2UtilResponse *idwsf2_util_response,
                                       const char *status);
```

Set the first level status code and no second level status code.

Parameters

idwsf2_util_response	a LassoIdWsf2UtilResponse object	
status	a status code identifier	

lasso_idwsf2_util_response_set_status2 ()

```
void
lasso_idwsf2_util_response_set_status2
                               (LassoIdWsf2UtilResponse *idwsf2_util_response,
                                const char *status,
                                const char *status2);
```

Types and Values

struct LassoIdWsf2UtilResponse

```
struct LassoIdWsf2UtilResponse {
  LassoNode parent;

  /* elements */
  LassoIdWsf2UtilStatus *Status;
  GList *Extension; /* of LassoIdWsf2Utilextension */
  /* attributes */
  char *itemIDRef;
  GHashTable *attributes;
};
```

11.133 LassoIdWsf2UtilStatus

LassoIdWsf2UtilStatus — <util:Status>

Functions

LassoIdWsf2UtilStatus *	lasso_idwsf2_util_status_new ()
LassoIdWsf2UtilStatus *	lasso_idwsf2_util_status_new_with_code ()

Types and Values

struct	LassoIdWsf2UtilStatus

Description

```
<xs:complexType name="StatusType">
  <xs:annotation>
    <xs:documentation>
      A type that may be used for status codes.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="Status" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="code" type="xs:string" use="required"/>
  <xs:attribute name="ref" type="IDReferenceType" use="optional"/>
  <xs:attribute name="comment" type="xs:string" use="optional"/>
</xs:complexType>
```

Figure 11.131: Schema fragment for util:Status

Functions

lasso_idwsf2_util_status_new ()

```
LassoIdWsf2UtilStatus~*
lasso_idwsf2_util_status_new (void);
```

Creates a new LassoIdWsf2UtilStatus object.

Returns

a newly created LassoIdWsf2UtilStatus object

lasso_idwsf2_util_status_new_with_code ()

```
LassoIdWsf2UtilStatus~*
lasso_idwsf2_util_status_new_with_code
                               (const gchar *code1,
                                const gchar *code2);
```

Creates a new LassoIdWsf2UtilStatus containing code1 and if code2 is not-NULL a nested LassoIdWsf2UtilStatus containing code2.

Parameters

code1	first level code
code2	second level code

Returns

a newly created LassoIdWsf2UtilStatus object

Types and Values

struct LassoIdWsf2UtilStatus

```
struct LassoIdWsf2UtilStatus {
  LassoNode parent;

  /* elements */
  GList *Status; /* of LassoIdWsf2UtilStatus */
  /* attributes */
  char *code;
  char *ref;
  char *comment;
};
```

Chapter 12

Object from the SOAP 1.1 schemas

12.1 LassoSoapBody

LassoSoapBody — <soap:Body>

Functions

LassoSoapBody *	lasso_soap_body_new ()
LassoSoapBody *	lasso_soap_body_new_from_message ()

Types and Values

struct	LassoSoapBody

Description

```
<xs:element name="Body" type="tns:Body"/>
  <xs:complexType name="Body">
  <xs:sequence>
    <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax">
<xs:annotation>
<xs:documentation>
    Prose in the spec does not specify that attributes are allowed on the Body element
</xs:documentation>
     </xs:annotation>
  </xs:anyAttribute>
</xs:complexType>
```

Figure 12.1: Schema fragment for soap:Body

Functions

lasso_soap_body_new ()

```
LassoSoapBody~*
lasso_soap_body_new (void);
```

lasso_soap_body_new_from_message ()

```
LassoSoapBody~*
lasso_soap_body_new_from_message (const gchar *message);
```

Types and Values

struct LassoSoapBody

```
struct LassoSoapBody {
  LassoNode parent;

  GList *any; /* of LassoNode */
  gchar *Id;
  /* TODO : supports of any attributes */
};
```

12.2 LassoSoapDetail

LassoSoapDetail —

Functions

LassoSoapDetail *	lasso_soap_detail_new ()
LassoSoapDetail *	lasso_soap_detail_new_from_message ()

Types and Values

struct	LassoSoapDetail

Description

Functions

lasso_soap_detail_new ()

```
LassoSoapDetail~*
lasso_soap_detail_new (void);
```

lasso_soap_detail_new_from_message ()

```
LassoSoapDetail~*
lasso_soap_detail_new_from_message (const gchar *message);
```

Types and Values

struct LassoSoapDetail

```
struct LassoSoapDetail {
  LassoNode parent;

  GList *any; /* of LassoNode */
};
```

12.3 LassoSoapEnvelope

LassoSoapEnvelope — <soap:Envelope>

Functions

LassoSoapEnvelope *	lasso_soap_envelope_new ()
LassoSoapEnvelope *	lasso_soap_envelope_new_full ()
LassoSoapEnvelope *	lasso_soap_envelope_new_from_message ()
void	lasso_soap_envelope_add_action ()
void	lasso_soap_envelope_add_security_token ()
void	lasso_soap_envelope_add_to_body ()
const char *	lasso_soap_envelope_get_action ()
GList *	lasso_soap_envelope_get_body_content ()
LassoWsAddrAttributedURI *	lasso_soap_envelope_get_message_id ()
LassoWsAddrRelatesTo *	lasso_soap_envelope_get_relates_to ()
LassoSaml2Assertion *	lasso_soap_envelope_get_saml2_security_token ()
LassoIdWsf2Sb2UserInteractionHint	lasso_soap_envelope_get_sb2_user_interaction_hint ()
char *	lasso_soap_envelope_sb2_get_provider_id ()
const char *	lasso_soap_envelope_sb2_get_redirect_request_url ()
LassoIdWsf2Sb2TargetIdentity *	lasso_soap_envelope_sb2_get_target_identity_header ()
void	lasso_soap_envelope_set_sb2_user_interaction_hint ()
LassoWsSec1SecurityHeader *	lasso_soap_envelope_wssec_get_security_header ()
LassoSoapFault *	lasso_soap_envelope_get_soap_fault ()
void	lasso_soap_envelope_set_relates_to ()

Types and Values

struct	LassoSoapEnvelope

Description

Functions

lasso_soap_envelope_new ()

```
LassoSoapEnvelope~*
lasso_soap_envelope_new (LassoSoapBody *body);
```

Creates a new LassoSoapEnvelope with a new empty LassoSoapBody member. Note, this function does not add a LassoSoapHeader, if you need both headers and a body use lasso_soap_envelope_new_full() instead.

Returns

new LassoSoapEnvelope

lasso_soap_envelope_new_full ()

```
LassoSoapEnvelope~*
lasso_soap_envelope_new_full (void);
```

Creates a new LassoSoapEnvelope with new empty LassoSoapHeader and LassoSoapBody members.

Returns

new LassoSoapEnvelope

lasso_soap_envelope_new_from_message ()

```
LassoSoapEnvelope~*
lasso_soap_envelope_new_from_message (const gchar *message);
```

Given an XML document in `message` , parse it and convert it into a LassoNode, then insert that LassoNode into the body of the newly returned LassoSoapEnvelope.

Parameters

message	XML document

Returns

new LassoSoapEnvelope

lasso_soap_envelope_add_action ()

```
void
lasso_soap_envelope_add_action (LassoSoapEnvelope *soap_envelope,
                                const char *action);
```

lasso_soap_envelope_add_security_token ()

```
void
lasso_soap_envelope_add_security_token
                               (LassoSoapEnvelope *soap_envelope,
                                LassoNode *token);
```

Add `token` as a security token to the headers of `soap_envelope` using a LassoWsSec1SecurityHeader element as a container. Eventually create the LassoWsSec1SecurityHeader element if it does not exist already.

Parameters

soap_envelope	a LassoSoapEnvelope object
token	a LassoNode to use as a security token

lasso_soap_envelope_add_to_body ()

```
void
lasso_soap_envelope_add_to_body (LassoSoapEnvelope *soap_envelope,
                                 LassoNode *content);
```

Add new content to the SOAP message body.

Parameters

soap_envelope	a LassoSoapEnvelope object
content	a LassoNode object to add to the body of the SOAP message

lasso_soap_envelope_get_action ()

```
const char~*
lasso_soap_envelope_get_action (LassoSoapEnvelope *soap_envelope);
```

lasso_soap_envelope_get_body_content ()

```
GList~*
lasso_soap_envelope_get_body_content (LassoSoapEnvelope *soap_envelope);
```

Return the body content of the `soap_envelope` object.

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

the content or NULL if the concent is empty or the object invalid.

[element-type LassoNode]

lasso_soap_envelope_get_message_id ()

```
LassoWsAddrAttributedURI~*
lasso_soap_envelope_get_message_id (LassoSoapEnvelope *soap_envelope,
                                    gboolean create);
```

Return the WS-Addressing header MessageID content.

Parameters

soap_envelope	a LassoSoapEnvelope object	
create	whether to create the node if it is not found.	*[default FALSE]*

Returns

a LassoWsAddrAttributedURI object or NULL if none is found, and creation was not allowed.

[transfer none]

lasso_soap_envelope_get_relates_to ()

```
LassoWsAddrRelatesTo~*
lasso_soap_envelope_get_relates_to (LassoSoapEnvelope *envelope,
                                    gboolean create);
```

Returns the RelatesTo WS-Addressing header, if it exists.

Parameters

soap_envelope	a LassoSoapEnvelope object	
create	whether to create the node if it is not found.	*[default FALSE]*

Returns

a LassoWsAddrAttributedURI object or NULL if none is found, and creation was not allowed.

[transfer none]

lasso_soap_envelope_get_saml2_security_token ()

```
LassoSaml2Assertion~*
lasso_soap_envelope_get_saml2_security_token
                               (LassoSoapEnvelope *soap_envelope);
```

Return a SAML2 Assertion used as a security token if one is found in the headers of `soap_envelope` .

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

a LassoSaml2Assertion, or NULL if none is found.

[transfer none]

lasso_soap_envelope_get_sb2_user_interaction_hint ()

```
LassoIdWsf2Sb2UserInteractionHint
lasso_soap_envelope_get_sb2_user_interaction_hint
                               (LassoSoapEnvelope *soap_envelope);
```

lasso_soap_envelope_sb2_get_provider_id ()

```
char~*
lasso_soap_envelope_sb2_get_provider_id
                               (LassoSoapEnvelope *soap_envelope);
```

Return the providerID contained in the sb2:Sender header.

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

the providerID string or NULL if no sb2:Sender header is present.

lasso_soap_envelope_sb2_get_redirect_request_url ()

```
const char~*
lasso_soap_envelope_sb2_get_redirect_request_url
                               (LassoSoapEnvelope *soap_envelope);
```

Return the redirect request URL from the sb2:RedirectRequest SOAP Fault detail.

The WSC MUST verify that this URL belong to the WSP. You can do this by comparing the domain with the one in the *LassoProfile.msg_url* field

The WSC MUST redirect the User Agent to this URL with a GET or POST request. It MUST add a parameter named ReturnToURL giving the URL where the WSP will send the User Agent after the interaction. It MAY add an IDP parameter indicating to the WSP how to authenticate the principal if no preexisting session with the User Agent exists

The WSP must check that the ReturnToURL belon to the WSP, by using the providerID URL for example.

After the interaction the WSC must redirect the User Agent to the ReturnToURL URL adding a parameter named ResendMessage. If ResendMessage is 0 or false, it means that the principal refused to continue the process. Any other value means that the prinicpal accepted and so the WSC can try again its request.

In order to succeed the request need to refer to the SOAP Fault response containing the RedirectRequest element. See lasso_soap_envelop and LassoWsAddrAttributedURI.

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

the redirect URL string or NULL if no sb2:RedirectRequest detail is present.

[transfer none][allow-none]

lasso_soap_envelope_sb2_get_target_identity_header ()

```
LassoIdWsf2Sb2TargetIdentity~*
lasso_soap_envelope_sb2_get_target_identity_header
                                (LassoSoapEnvelope *soap_envelope);
```

Return the first sb2:TargetIdentity header.

Parameters

soap_envelope	a LassoSoapEnvelope object.

Returns

the first LassoIdWsf2Sb2TargetIdentity object found in the headers of the `soap_envelope` , or NULL if none is found.

[transfer none]

lasso_soap_envelope_set_sb2_user_interaction_hint ()

```
void
lasso_soap_envelope_set_sb2_user_interaction_hint
                                (LassoSoapEnvelope *soap_envelope,
                                 LassoIdWsf2Sb2UserInteractionHint hint);
```

lasso_soap_envelope_wssec_get_security_header ()

```
LassoWsSec1SecurityHeader~*
lasso_soap_envelope_wssec_get_security_header
                                (LassoSoapEnvelope *soap_envelope);
```

Return the first wsse:Security header found in the headers of `soap_envelope` .

Parameters

soap_envelope	a LassoSoapEnvelope object

Returns

the wsse:Security header found in the SOAP message, or NULL if none is found.

[transfer none]

lasso_soap_envelope_get_soap_fault ()

```
LassoSoapFault~*
lasso_soap_envelope_get_soap_fault (LassoSoapEnvelope *soap_envelope,
                                    gboolean create);
```

Return the first SOAP Fault in the Body of the soap message *soap_envelope* .

Parameters

soap_envelope	a LassoSoapEnvelope	
create	whether to create the SOAP Fault.	*[default FALSE]*

Returns

a LassoSoapFault object or NULL.

[transfer none][allow-none]

lasso_soap_envelope_set_relates_to ()

```
void
lasso_soap_envelope_set_relates_to (LassoSoapEnvelope *soap_envelope,
                                    char *value,
                                    char *relationship);
```

Set the value of the RelatesTo header, if it does not exist it is created. If *value* is NULL, the header is removed.

Parameters

soap_envelope	a LassoSoapEnvelope object	
value	the value to set into the new header.	*[allow-none]*
relationship	kind of the relationship.	*[allow-none]*

Types and Values

struct LassoSoapEnvelope

```
struct LassoSoapEnvelope {
  LassoNode parent;

  LassoSoapHeader *Header;
  LassoSoapBody *Body;
};
```

12.4 LassoSoapFault

LassoSoapFault — object mapping for a soap11:Fault

Stability Level

Stable, unless otherwise indicated

Functions

LassoSoapFault *	lasso_soap_fault_new ()
LassoSoapFault *	lasso_soap_fault_new_from_message ()
LassoSoapFault *	lasso_soap_fault_new_full ()
void	lasso_soap_fault_add_to_detail ()
LassoSoapDetail *	lasso_soap_fault_get_detail ()

Types and Values

struct	LassoSoapFault

Description

```
<xs: element name="Fault" type="tns: Fault"/>
<xs: complexType name="Fault" final="extension">
  <xs: annotation>
    <xs: documentation>
      Fault reporting structure
    </xs: documentation>
  </xs: annotation>
  <xs: sequence>
    <xs: element name="faultcode" type="xs: QName"/>
    <xs: element name="faultstring" type="xs: string"/>
    <xs: element name="faultactor" type="xs: anyURI" minOccurs="0"/>
    <xs: element name="detail" type="tns: detail" minOccurs="0"/>
  </xs: sequence>
 </xs: complexType>

 <xs: complexType name="detail">
   <xs: sequence>
     <xs: any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
   </xs: sequence>
   <xs: anyAttribute namespace="##any" processContents="lax"/>
 </xs: complexType>
```

Figure 12.2: Schema fragment for soap11:Fault

Functions

lasso_soap_fault_new ()

```
LassoSoapFault~*
lasso_soap_fault_new (void);
```

lasso_soap_fault_new_from_message ()

```
LassoSoapFault~*
lasso_soap_fault_new_from_message (const gchar *message);
```

lasso_soap_fault_new_full ()

```
LassoSoapFault~*
lasso_soap_fault_new_full (const char *faultcode,
                           const char *faultstring);
```

lasso_soap_fault_add_to_detail ()

```
void
lasso_soap_fault_add_to_detail (LassoSoapFault *soap_fault,
                                LassoNode *node);
```

Fill the detail part of a SOAP fault.

Parameters

soap_fault	a LassoSoapFault object
node	a node to add to the Detail sub-element of the *soap_fault* object.

lasso_soap_fault_get_detail ()

```
LassoSoapDetail~*
lasso_soap_fault_get_detail (LassoSoapFault *soap_fault,
                             gboolean create);
```

Types and Values

struct LassoSoapFault

```
struct LassoSoapFault {
  LassoNode parent;

  gchar *faultcode;
  gchar *faultstring;
  GList *faultactor; /* of string */
  LassoSoapDetail *Detail;
};
```

12.5 LassoSoapHeader

LassoSoapHeader — <soap:Header>

Functions

LassoSoapHeader *	lasso_soap_header_new ()
LassoSoapHeader *	lasso_soap_header_new_from_message ()

Types and Values

struct	LassoSoapHeader

Description

Functions

lasso_soap_header_new ()

```
LassoSoapHeader~*
lasso_soap_header_new (void);
```

lasso_soap_header_new_from_message ()

```
LassoSoapHeader~*
lasso_soap_header_new_from_message (const gchar *message);
```

Types and Values

struct LassoSoapHeader

```
struct LassoSoapHeader {
  LassoNode parent;

  GList *Other; /* of LassoNode */
};
```

12.6 Strings for SOAP

Strings for SOAP —

Types and Values

#define	LASSO_SOAP_ENV_HREF
#define	LASSO_SOAP_ENV_PREFIX
#define	LASSO_SOAP_ENV_ACTOR

Description

Functions

Types and Values

LASSO_SOAP_ENV_HREF

```
#define LASSO_SOAP_ENV_HREF   "http://schemas.xmlsoap.org/soap/envelope/"
```

Namespace for SOAP 1.1 messages

LASSO_SOAP_ENV_PREFIX

```
#define LASSO_SOAP_ENV_PREFIX "s"
```

Preferred prefix for namespace of SOAP 1.1 messages

LASSO_SOAP_ENV_ACTOR

```
#define LASSO_SOAP_ENV_ACTOR "http://schemas.xmlsoap.org/soap/actor/next"
```

Chapter 13

Object from the XML-DSIG schemas

13.1 LassoDsKeyInfo

LassoDsKeyInfo — object mapping for an XML DSIG KeyInfo element

Functions

LassoDsKeyInfo *	lasso_ds_key_info_new ()

Types and Values

struct	LassoDsKeyInfo

Description

Functions

lasso_ds_key_info_new ()

```
LassoDsKeyInfo~*
lasso_ds_key_info_new (void);
```

Creates a new LassoDsKeyInfo object.

Returns

a newly created LassoDsKeyInfo object

Types and Values

struct LassoDsKeyInfo

```
struct LassoDsKeyInfo {
  LassoNode parent;

  LassoDsKeyValue *KeyValue;
};
```

13.2 LassoDsKeyValue

LassoDsKeyValue — object mapping for an XML DSIG KeyValue element

Functions

LassoDsKeyValue *	lasso_ds_key_value_new ()

Types and Values

struct	LassoDsKeyValue

Description

Functions

lasso_ds_key_value_new ()

```
LassoDsKeyValue~*
lasso_ds_key_value_new (void);
```

Creates a new LassoDsKeyValue object.

Returns

a newly created LassoDsKeyValue object

Types and Values

struct LassoDsKeyValue

```
struct LassoDsKeyValue {
  LassoNode parent;

  LassoDsRsaKeyValue *RSAKeyValue;
};
```

13.3 LassoDsRsaKeyValue

LassoDsRsaKeyValue —

Functions

LassoDsRsaKeyValue *	lasso_ds_rsa_key_value_new ()

Types and Values

struct	LassoDsRsaKeyValue

Description

Functions

lasso_ds_rsa_key_value_new ()

```
LassoDsRsaKeyValue~*
lasso_ds_rsa_key_value_new (void);
```

Creates a new LassoDsRsaKeyValue object.

Returns

a newly created LassoDsRsaKeyValue object

Types and Values

struct LassoDsRsaKeyValue

```
struct LassoDsRsaKeyValue {
  LassoNode parent;

  char *Modulus;
  char *Exponent;
};
```

13.4 Strings for XML-DSIG

Strings for XML-DSIG — General strings constants for XMLDsig

Types and Values

#define	LASSO_DS_HREF
#define	LASSO_DS_PREFIX

Includes

```
#include <lasso/xml/dsig/strings.h>
```

Description

Functions

Types and Values

LASSO_DS_HREF

```
#define LASSO_DS_HREF   "http://www.w3.org/2000/09/xmldsig#"
```

Namespace for FIXME

LASSO_DS_PREFIX

```
#define LASSO_DS_PREFIX "ds"
```

Preferred prefix for namespace of FIXME

Chapter 14

Object from the WS-* schemas

14.1 LassoWsAddrAttributedAny

LassoWsAddrAttributedAny —

Functions

LassoWsAddrAttributedAny *	lasso_wsa_attributed_any_new ()

Types and Values

struct	LassoWsAddrAttributedAny

Description

Functions

lasso_wsa_attributed_any_new ()

```
LassoWsAddrAttributedAny~*
lasso_wsa_attributed_any_new (void);
```

Creates a new LassoWsAddrAttributedAny object.

Returns

a newly created LassoWsAddrAttributedAny object

Types and Values

struct LassoWsAddrAttributedAny

```
struct LassoWsAddrAttributedAny {
  LassoNode parent;
  /* elements */
  GList *any; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

14.2 LassoWsAddrAttributedQName

LassoWsAddrAttributedQName —

Functions

LassoWsAddrAttributedQName *	lasso_wsa_attributed_qname_new ()
LassoWsAddrAttributedQName *	lasso_wsa_attributed_qname_new_with_string ()

Types and Values

struct	LassoWsAddrAttributedQName

Description

Functions

lasso_wsa_attributed_qname_new ()

```
LassoWsAddrAttributedQName~*
lasso_wsa_attributed_qname_new (void);
```

Creates a new LassoWsAddrAttributedQName object.

Returns

a newly created LassoWsAddrAttributedQName object

lasso_wsa_attributed_qname_new_with_string ()

```
LassoWsAddrAttributedQName~*
lasso_wsa_attributed_qname_new_with_string
                               (char *content);
```

Creates a new LassoWsAddrAttributedQName object and initializes it with `content` as content.

Parameters

content	a content string.	

Returns

a newly created LassoWsAddrAttributedQName object

Types and Values

struct LassoWsAddrAttributedQName

```
struct LassoWsAddrAttributedQName {
  LassoNode parent;

  /* elements */
  char *content;
  /* attributes */
  GHashTable *attributes;
};
```

14.3 LassoWsAddrAttributedUnsignedLong

LassoWsAddrAttributedUnsignedLong —

Functions

LassoWsAddrAttributedUnsignedLong *	lasso_wsa_attributed_unsigned_long_new ()

Types and Values

struct	LassoWsAddrAttributedUnsignedLong

Description

Functions

lasso_wsa_attributed_unsigned_long_new ()

```
LassoWsAddrAttributedUnsignedLong~*
lasso_wsa_attributed_unsigned_long_new
                               (void);
```

Creates a new LassoWsAddrAttributedUnsignedLong object.

Returns

a newly created LassoWsAddrAttributedUnsignedLong object

Types and Values

struct LassoWsAddrAttributedUnsignedLong

```
struct LassoWsAddrAttributedUnsignedLong {
  LassoNode parent;

  /* elements */
  int content;
  /* attributes */
  GHashTable *attributes;
};
```

14.4 LassoWsAddrAttributedURI

LassoWsAddrAttributedURI —

Functions

LassoWsAddrAttributedURI *	lasso_wsa_attributed_uri_new ()
LassoWsAddrAttributedURI *	lasso_wsa_attributed_uri_new_with_string ()

Types and Values

struct	LassoWsAddrAttributedURI

Description

Functions

lasso_wsa_attributed_uri_new ()

```
LassoWsAddrAttributedURI~*
lasso_wsa_attributed_uri_new (void);
```

Creates a new LassoWsAddrAttributedURI object.

Returns

a newly created LassoWsAddrAttributedURI object

lasso_wsa_attributed_uri_new_with_string ()

```
LassoWsAddrAttributedURI~*
lasso_wsa_attributed_uri_new_with_string
                              (const char *content);
```

Creates a new LassoWsAddrAttributedURI object and initializes it with `content` as content.

Parameters

content	a content string	

Returns

a newly created LassoWsAddrAttributedURI object

Types and Values

struct LassoWsAddrAttributedURI

```
struct LassoWsAddrAttributedURI {
  LassoNode parent;

  /* elements */
  char *content;
  /* attributes */
  GHashTable *attributes;
};
```

14.5 LassoWsAddrEndpointReference

LassoWsAddrEndpointReference —

Functions

LassoWsAddrEndpointReference *	lasso_wsa_endpoint_reference_new ()
LassoNode *	lasso_wsa_endpoint_reference_get_target_identity_token ()
LassoWsAddrEndpointReference *	lasso_wsa_endpoint_reference_new_for_idwsf2_service ()
lasso_error_t	lasso_wsa_endpoint_reference_add_security_token ()
lasso_error_t	lasso_wsa_endpoint_reference_associate_service_to_type ()
LassoIdWsf2Profile *	lasso_wsa_endpoint_reference_get_service ()

Types and Values

struct	LassoWsAddrEndpointReference

Description

Functions

lasso_wsa_endpoint_reference_new ()

```
LassoWsAddrEndpointReference~*
lasso_wsa_endpoint_reference_new (void);
```

Creates a new LassoWsAddrEndpointReference object.

Returns

a newly created LassoWsAddrEndpointReference object

lasso_wsa_endpoint_reference_get_target_identity_token ()

```
LassoNode~*
lasso_wsa_endpoint_reference_get_target_identity_token
                               (const LassoWsAddrEndpointReference *epr,
                                gboolean (*sech_mech_predicate) (const char *),
                                const char *security_mech_id);
```

Return the first target identity token found in the metadata of the *epr* object which qualify with respect to the predicate or the given security mechanism identifier. It is an error to pass both of *sech_mech_predicate* and *security_mech_id* as NULL.

Parameters

epr	a LassoWsAddrEndpointReference object	
sech_mech_predicate	a boolean function to select the security mechanism for which we want the security token.	*[allow-none]*
security_mech_id	an optional specific security mechanism identifier to select the security token.	*[allow-none]*

Returns

a LassoNode object or NULL if the query cannot be satisfied.

[transfer none]

lasso_wsa_endpoint_reference_new_for_idwsf2_service ()

```
LassoWsAddrEndpointReference~*
lasso_wsa_endpoint_reference_new_for_idwsf2_service
                             (const char *address,
                              const char *service_ype,
                              const char *provider_id,
                              const char *abstract);
```

Create and populate a new LassoWsAddrEndpointReference object.

Parameters

address	the URL of the SOAP endpoint where the service is anchored	
service_type	an URI identifying the ID-WSF 2.0 service type	
provider_id	an URI identifying the SAML 2.0 service provider hosting the service, this should help in finding key material for security mechanisms.	
abstract	a human description of the service.	

Returns

a newly created LassoWsAddrEndpointReference.

lasso_wsa_endpoint_reference_add_security_token ()

```
lasso_error_t
lasso_wsa_endpoint_reference_add_security_token
                               (LassoWsAddrEndpointReference *epr,
                                LassoNode *security_token,
                                GList *security_mechanisms);
```

Add a new security context declaration for the given security mechanisms identifiers and populate it with a security token.

Parameters

epr	a LassoWsAddrEndpointReference object	
security_token	a security token as a LassoNode object	
security_mechanisms	a list of security mechanism for whom the token is made.	*[element-type utf8]*

Returns

0 if successfull, an error code otherwise.

lasso_wsa_endpoint_reference_associate_service_to_type ()

```
lasso_error_t
lasso_wsa_endpoint_reference_associate_service_to_type
                               (const char *service_type_uri,
                                GType g_type);
```

Associate a profile type to a service type.

Parameters

service_type_uri	a service type to associate	
g_type	the type of the profile object handling this service type	

Returns

0 if successful, an error code otherwise.

lasso_wsa_endpoint_reference_get_service ()

```
LassoIdWsf2Profile~*
lasso_wsa_endpoint_reference_get_service
                               (LassoWsAddrEndpointReference *epr);
```

Get a profile object able to communicate with the service represented by this EPR.

Return object: a newly created LassoIdWsf2Profile instance.

Parameters

epr	a LassoWsAddrEndpointReference object

Types and Values

struct LassoWsAddrEndpointReference

```
struct LassoWsAddrEndpointReference {
  LassoNode parent;

  /* elements */
  LassoWsAddrAttributedURI *Address;
  LassoWsAddrReferenceParameters *ReferenceParameters;
  LassoWsAddrMetadata *Metadata;
  GList *any; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

14.6 LassoWsAddrMetadata

LassoWsAddrMetadata —

Functions

LassoWsAddrMetadata *	lasso_wsa_metadata_new ()

Types and Values

struct	LassoWsAddrMetadata

Description

Functions

lasso_wsa_metadata_new ()

```
LassoWsAddrMetadata~*
lasso_wsa_metadata_new (void);
```

Creates a new LassoWsAddrMetadata object.

Returns

a newly created LassoWsAddrMetadata object

Types and Values

struct LassoWsAddrMetadata

```
struct LassoWsAddrMetadata {
  LassoNode parent;

  /* elements */
  GList *any; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

14.7 LassoWsAddrProblemAction

LassoWsAddrProblemAction —

Functions

LassoWsAddrProblemAction *	lasso_wsa_problem_action_new ()

Types and Values

struct	LassoWsAddrProblemAction

Description

Functions

lasso_wsa_problem_action_new ()

```
LassoWsAddrProblemAction~*
lasso_wsa_problem_action_new (void);
```

Creates a new LassoWsAddrProblemAction object.

Returns

a newly created LassoWsAddrProblemAction object

Types and Values

struct LassoWsAddrProblemAction

```
struct LassoWsAddrProblemAction {
  LassoNode parent;

  /* elements */
  LassoWsAddrAttributedURI *Action;
  char *SoapAction;
  /* attributes */
  GHashTable *attributes;
};
```

14.8 LassoWsAddrReferenceParameters

LassoWsAddrReferenceParameters —

Functions

LassoWsAddrReferenceParameters *	lasso_wsa_reference_parameters_new ()

Types and Values

struct	LassoWsAddrReferenceParameters

Description

Functions

lasso_wsa_reference_parameters_new ()

```
LassoWsAddrReferenceParameters~*
lasso_wsa_reference_parameters_new (void);
```

Creates a new LassoWsAddrReferenceParameters object.

Returns

a newly created LassoWsAddrReferenceParameters object

Types and Values

struct LassoWsAddrReferenceParameters

```
struct LassoWsAddrReferenceParameters {
  LassoNode parent;

  /* elements */
  GList *any; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

14.9 LassoWsAddrRelatesTo

LassoWsAddrRelatesTo —

Functions

LassoWsAddrRelatesTo *	lasso_wsa_relates_to_new ()
LassoWsAddrRelatesTo *	lasso_wsa_relates_to_new_with_string ()

Types and Values

struct	LassoWsAddrRelatesTo

Description

Functions

lasso_wsa_relates_to_new ()

```
LassoWsAddrRelatesTo~*
lasso_wsa_relates_to_new (void);
```

Creates a new LassoWsAddrRelatesTo object.

Returns

a newly created LassoWsAddrRelatesTo object

lasso_wsa_relates_to_new_with_string ()

```
LassoWsAddrRelatesTo~*
lasso_wsa_relates_to_new_with_string (char *content);
```

Creates a new LassoWsAddrRelatesTo object and initializes it with `content` as content.

Parameters

content	a content string

Returns

a newly created LassoWsAddrRelatesTo object

Types and Values

struct LassoWsAddrRelatesTo

```
struct LassoWsAddrRelatesTo {
  LassoNode parent;
  /* elements */
  char *content;
  /* attributes */
  char *RelationshipType;
  GHashTable *attributes;
};
```

14.10 LassoWsSec1Embedded

LassoWsSec1Embedded —

Functions

LassoWsSec1Embedded *	lasso_wsse_embedded_new ()

Types and Values

struct	LassoWsSec1Embedded

Description

Functions

lasso_wsse_embedded_new ()

```
LassoWsSec1Embedded~*
lasso_wsse_embedded_new (void);
```

Creates a new LassoWsSec1Embedded object.

Returns

a newly created LassoWsSec1Embedded object

Types and Values

struct LassoWsSec1Embedded

```
struct LassoWsSec1Embedded {
  LassoNode parent;

  /* attributes */
  char *ValueType;
  GHashTable *attributes;
};
```

14.11 LassoWsSec1Reference

LassoWsSec1Reference —

Functions

LassoWsSec1Reference *	lasso_wsse_reference_new ()

Types and Values

struct	LassoWsSec1Reference

Description

Functions

lasso_wsse_reference_new ()

```
LassoWsSec1Reference~*
lasso_wsse_reference_new (void);
```

Creates a new LassoWsSec1Reference object.

Returns

a newly created LassoWsSec1Reference object

Types and Values

struct LassoWsSec1Reference

```
struct LassoWsSec1Reference {
  LassoNode parent;

  /* attributes */
  char *URI;
  char *ValueType;
  GHashTable *attributes;
};
```

14.12 LassoWsSec1SecurityHeader

LassoWsSec1SecurityHeader —

Functions

LassoWsSec1SecurityHeader *	lasso_wsse_security_header_new ()

Types and Values

struct	LassoWsSec1SecurityHeader

Description

Functions

lasso_wsse_security_header_new ()

```
LassoWsSec1SecurityHeader~*
lasso_wsse_security_header_new (void);
```

Creates a new LassoWsSec1SecurityHeader object.

Returns

a newly created LassoWsSec1SecurityHeader object

Types and Values

struct LassoWsSec1SecurityHeader

```
struct LassoWsSec1SecurityHeader {
  LassoNode parent;

  /* elements */
  GList *any; /* of LassoNode */
  /* attributes */
  GHashTable *attributes;
};
```

14.13 LassoWsSec1SecurityTokenReference

LassoWsSec1SecurityTokenReference —

Functions

LassoWsSec1SecurityTokenReference	* lasso_wsse_security_token_reference_new ()

Types and Values

struct	LassoWsSec1SecurityTokenReference

Description

Functions

lasso_wsse_security_token_reference_new ()

```
LassoWsSec1SecurityTokenReference~*
lasso_wsse_security_token_reference_new
                               (void);
```

Creates a new LassoWsSec1SecurityTokenReference object.

Returns

a newly created LassoWsSec1SecurityTokenReference object

Types and Values

struct LassoWsSec1SecurityTokenReference

```
struct LassoWsSec1SecurityTokenReference {
  LassoNode parent;

  /* attributes */
  char *Id;
  char *Usage;
  GHashTable *attributes;
};
```

14.14 LassoWsSec1TransformationParameters

LassoWsSec1TransformationParameters —

Functions

LassoWsSec1TransformationParameters *	lasso_wsse_transformation_parameters_new ()

Types and Values

struct	LassoWsSec1TransformationParameters

Description

Functions

lasso_wsse_transformation_parameters_new ()

```
LassoWsSec1TransformationParameters~*
lasso_wsse_transformation_parameters_new
                               (void);
```

Creates a new LassoWsSec1TransformationParameters object.

Returns

a newly created LassoWsSec1TransformationParameters object

Types and Values

struct LassoWsSec1TransformationParameters

```
struct LassoWsSec1TransformationParameters {
  LassoNode parent;

  /* attributes */
  GHashTable *attributes;
};
```

14.15 LassoWsseUsernameToken

LassoWsseUsernameToken —

Functions

LassoWsseUsernameToken *	lasso_wsse_username_token_new ()
void	lasso_wsse_username_token_reset_nonce ()
void	lasso_wsse_username_token_set_password_kind ()
lasso_error_t	lasso_wsse_username_token_set_password ()
lasso_error_t	lasso_wsse_username_token_check_password ()
guchar *	lasso_wsse_username_token_derive_key ()

Types and Values

enum	LassoWsseUsernameTokenPasswordType
struct	LassoWsseUsernameToken

Description

Transmit username and password credential as a WS-Security token. The password can be transmitted as cleartext or using a digest mode. It also allows to derive encryption and HMAC signing keys.

Functions

lasso_wsse_username_token_new ()

```
LassoWsseUsernameToken~*
lasso_wsse_username_token_new (void);
```

lasso_wsse_username_token_reset_nonce ()

```
void
lasso_wsse_username_token_reset_nonce (LassoWsseUsernameToken *wsse_username_token);
```

Generate a random nonce.

Parameters

wsse_username_token	a LassoWsseUsernameToken object

lasso_wsse_username_token_set_password_kind ()

```
void
lasso_wsse_username_token_set_password_kind
                               (LassoWsseUsernameToken *wsse_username_token,
                                LassoWsseUsernameTokenPasswordType password_type);
```

Set the way to transmit password, that is either cleartext or digest.

Parameters

wsse_username_token	a LassoWsseUsernameToken object
password_type	a LassoWsseUsernameTokenPasswordType enumeration

lasso_wsse_username_token_set_password ()

```
lasso_error_t
lasso_wsse_username_token_set_password
                               (LassoWsseUsernameToken *wsse_username_token,
                                char *password);
```

Set the password using the given UTF-8 string. If password kind is digest, compute the digest SHA1(nonce + created + password), convert to Base64 and set it as the password. If nonce or created are NULL, the empty string is used.

Parameters

wsse_username_token	a LassoWsseUsernameToken object
password	an UTF-8 string

Returns

0 if successfull, an error code otherwise.

lasso_wsse_username_token_check_password ()

```
lasso_error_t
lasso_wsse_username_token_check_password
                               (LassoWsseUsernameToken *wsse_username_token,
                                char *password);
```

lasso_wsse_username_token_derive_key ()

```
guchar~*
lasso_wsse_username_token_derive_key (LassoWsseUsernameToken *wsse_username_token,
                                      char *password);
```

Generate a derived 128bit key using the password and setting from the UsernameToken.

Parameters

wsse_username_token	a LassoWsseUsernameToken object	
password	the known password	

Returns

a 20 byte octet string.

Types and Values

enum LassoWsseUsernameTokenPasswordType

Members

LASSO_WSSE_USERNAME_TOKEN_PASSWORD_TYPE_UNKNOWN		
LASSO_WSSE_USERNAME_TOKEN_PASSWORD_TYPE_TEXT		
LASSO_WSSE_USERNAME_TOKEN_PASSWORD_TYPE_DIGEST		
LASSO_WSSE_USERNAME_TOKEN_PASSWORD_TYPE_LAST		

struct LassoWsseUsernameToken

```
struct LassoWsseUsernameToken {
  LassoNode parent;

  char *Id;
  char *Username;
  char *Nonce;
  char *Salt;
  char *Created;
  int Iteration;
  GHashTable *attributes;
};
```

Members

LassoNode *parent*;		

char **Id*;	the identifier of the UsernameToken
char **Username*;	the username
char **Nonce*;	a nonce used to compute the digest of the password
char **Salt*;	the salt for generating derived key

char **Created*;	the timestamp for the generation of the token, also used in the digest of the password	
int *Iteration*;	how many times to apply SHA1 for generating derivated key	
GHashTable **attributes*;		

14.16 LassoWsUtil1Timestamp

LassoWsUtil1Timestamp —

Functions

LassoWsUtil1Timestamp *	lasso_wsu_timestamp_new ()

Types and Values

struct	LassoWsUtil1Timestamp

Description

Functions

lasso_wsu_timestamp_new ()

```
LassoWsUtil1Timestamp~*
lasso_wsu_timestamp_new (void);
```

Creates a new LassoWsUtil1Timestamp object.

Returns

a newly created LassoWsUtil1Timestamp object

Types and Values

struct LassoWsUtil1Timestamp

```
struct LassoWsUtil1Timestamp {
  LassoNode parent;

  /* elements */
  char *Created;
  char *Expires;
  /* attributes */
  char *Id;
  GHashTable *attributes;
};
```

14.17 Strings for WS-*

Strings for WS-* —

Types and Values

#define	LASSO_WSSE_HREF
#define	LASSO_WSSE_PREFIX
#define	LASSO_WSSE1_HREF
#define	LASSO_WSSE1_PREFIX
#define	LASSO_WSUTIL1_HREF
#define	LASSO_WSUTIL1_PREFIX
#define	LASSO_WSA_HREF
#define	LASSO_WSA_PREFIX
#define	LASSO_WSU_HREF
#define	LASSO_WSU_PREFIX
#define	LASSO_XSI_HREF
#define	LASSO_XSI_PREFIX

Description

Functions

Types and Values

LASSO_WSSE_HREF

```
#define LASSO_WSSE_HREF "http://schemas.xmlsoap.org/ws/2002/07/secext"
```

Namespace for FIXME

LASSO_WSSE_PREFIX

```
#define LASSO_WSSE_PREFIX "wsse"
```

Preferred prefix for namespace of FIXME

LASSO_WSSE1_HREF

```
#define             LASSO_WSSE1_HREF
```

Namespace for WS-Security 1.0

LASSO_WSSE1_PREFIX

```
#define LASSO_WSSE1_PREFIX "wsse"
```

Preferred prefix for namespace of WS-Security 1.0

LASSO_WSUTIL1_HREF

```
#define             LASSO_WSUTIL1_HREF
```

Namespace for FIXME

LASSO_WSUTIL1_PREFIX

```
#define LASSO_WSUTIL1_PREFIX "wsutil"
```

Preferred prefix for namespace of FIXME

LASSO_WSA_HREF

```
#define LASSO_WSA_HREF "http://www.w3.org/2005/08/addressing"
```

Namespace for FIXME

LASSO_WSA_PREFIX

```
#define LASSO_WSA_PREFIX "wsa"
```

Preferred prefix for namespace of FIXME

LASSO_WSU_HREF

```
#define             LASSO_WSU_HREF
```

Namespace for FIXME

LASSO_WSU_PREFIX

```
#define LASSO_WSU_PREFIX "wsu"
```

Preferred prefix for namespace of FIXME

LASSO_XSI_HREF

```
#define LASSO_XSI_HREF "http://www.w3.org/2001/XMLSchema-instance"
```

Namespace for XMLSchema instance

LASSO_XSI_PREFIX

```
#define LASSO_XSI_PREFIX "xsi"
```

Preferred prefix for namespace of XMLSChema instance

Chapter 15

Annotation Glossary

A

allow-none
NULL is OK, both for passing and for returning.

D

default
Default parameter value (in case a function which shadows this one via rename-to has fewer parameters).

E

element-type
Generics and defining elements of containers and arrays.

O

optional
NULL may be passed instead of a pointer to a location.

out
Parameter for returning results. Default is transfer full.

P

Private
An interface that can be used within the GNOME stack itself, but that is not documented for end-users. Such functions should only be used in specified and documented ways.

S

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.

U

Unstable

Unstable interfaces are experimental or transitional. They are typically used to give outside developers early access to new or rapidly changing technology, or to provide an interim solution to a problem where a more general solution is anticipated. No claims are made about either source or binary compatibility from one minor release to the next. The Unstable interface level is a warning that these interfaces are subject to change without warning and should not be used in unbundled products. Given such caveats, customer impact need not be a factor when considering incompatible changes to an Unstable interface in a major or minor release. Nonetheless, when such changes are introduced, the changes should still be mentioned in the release notes for the affected release.

Chapter 16

Index