

Hosting Environment (Daemon) Chain Components Reference Manual

Generated by Doxygen 1.4.7

Wed Feb 23 01:10:51 2011

Contents

1	Hosting Environment (Daemon) Chain Components Namespace Index	1
1.1	Hosting Environment (Daemon) Chain Components Namespace List	1
2	Hosting Environment (Daemon) Chain Components Hierarchical Index	3
2.1	Hosting Environment (Daemon) Chain Components Class Hierarchy	3
3	Hosting Environment (Daemon) Chain Components Data Structure Index	5
3.1	Hosting Environment (Daemon) Chain Components Data Structures	5
4	Hosting Environment (Daemon) Chain Components Namespace Documentation	7
4.1	ArcSec Namespace Reference	7
5	Hosting Environment (Daemon) Chain Components Data Structure Documentation	11
5.1	ArcSec::AllowPDP Class Reference	11
5.2	ArcSec::ArcAlgFactory Class Reference	12
5.3	ArcSec::ArcAttributeFactory Class Reference	13
5.4	ArcSec::ArcAttributeProxy< TheAttribute > Class Template Reference	14
5.5	ArcSec::ArcAuthZ Class Reference	15
5.6	ArcSec::ArcEvaluationCtx Class Reference	16
5.7	ArcSec::ArcEvaluator Class Reference	17
5.8	ArcSec::ArcFnFactory Class Reference	18
5.9	ArcSec::ArcPDP Class Reference	19
5.10	ArcSec::ArcPolicy Class Reference	20
5.11	ArcSec::ArcRequestItem Class Reference	21
5.12	ArcSec::ArcRequestTuple Class Reference	22
5.13	ArcSec::ArcRule Class Reference	23
5.14	ArcSec::DelegationPDP Class Reference	24
5.15	ArcSec::DenyPDP Class Reference	25
5.16	Arc::LDAPQuery Class Reference	26
5.17	Arc::MCC_HTTP Class Reference	27

5.18 Arc::MCC_HTTP_Client Class Reference	28
5.19 Arc::MCC_HTTP_Service Class Reference	29
5.20 Arc::MCC_SOAP Class Reference	30
5.21 Arc::MCC_SOAP_Service Class Reference	31
5.22 Arc::MCC_TCP Class Reference	32
5.23 Arc::MCC_TCP_Client Class Reference	33
5.24 Arc::MCC_TCP_Service Class Reference	34
5.25 Arc::MCC_TLS Class Reference	35
5.26 Arc::MCC_TLS_Client Class Reference	36
5.27 Arc::MCC_TLS_Service Class Reference	37
5.28 Arc::PayloadHTTP Class Reference	38
5.29 Arc::PayloadTCPSocket Class Reference	43
5.30 Arc::PayloadTLSStream Class Reference	44
5.31 ArcSec::PDPServiceInvoker Class Reference	46
5.32 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference	47
5.33 ArcSec::SAMLTokenSH Class Reference	48
5.34 ArcSec::SimpleListPDP Class Reference	49
5.35 Arc::SRMClient Class Reference	50
5.36 Arc::SRMClientRequest Class Reference	59
5.37 SRMFileInfo Class Reference	62
5.38 Arc::SRMFileMetaData Struct Reference	63
5.39 SRMInfo Class Reference	64
5.40 ArcSec::UsernameTokenSH Class Reference	65
5.41 ArcSec::X509TokenSH Class Reference	66
5.42 ArcSec::XACMLAlgFactory Class Reference	67
5.43 ArcSec::XACMLAttributeFactory Class Reference	68
5.44 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference	69
5.45 ArcSec::XACMLCondition Class Reference	70
5.46 ArcSec::XACMLEvaluationCtx Class Reference	71
5.47 ArcSec::XACMLEvaluator Class Reference	72
5.48 ArcSec::XACMLFnFactory Class Reference	73
5.49 ArcSec::XACMLPDP Class Reference	74
5.50 ArcSec::XACMLPolicy Class Reference	75
5.51 ArcSec::XACMLRule Class Reference	76
5.52 ArcSec::XACMLTarget Class Reference	77

Chapter 1

Hosting Environment (Daemon) Chain Components Namespace Index

1.1 Hosting Environment (Daemon) Chain Components Namespace List

Here is a list of all documented namespaces with brief descriptions:

ArcSec (ArcRequest, Parsing the specified Arc request format)	7
---	----------

Chapter 2

Hosting Environment (Daemon) Chain Components Hierarchical Index

2.1 Hosting Environment (Daemon) Chain Components Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ArcSec::AllowPDP	11
ArcSec::ArcAlgFactory	12
ArcSec::ArcAttributeFactory	13
ArcSec::ArcAttributeProxy< TheAttribute >	14
ArcSec::ArcAuthZ	15
ArcSec::ArcEvaluationCtx	16
ArcSec::ArcEvaluator	17
ArcSec::ArcFnFactory	18
ArcSec::ArcPDP	19
ArcSec::ArcPolicy	20
ArcSec::ArcRequestItem	21
ArcSec::ArcRequestTuple	22
ArcSec::ArcRule	23
ArcSec::DelegationPDP	24
ArcSec::DenyPDP	25
Arc::LDAPQuery	26
Arc::MCC_HTTP	27
Arc::MCC_HTTP_Client	28
Arc::MCC_HTTP_Service	29
Arc::MCC_SOAP	30
Arc::MCC_SOAP_Service	31
Arc::MCC_TCP	32
Arc::MCC_TCP_Client	33
Arc::MCC_TCP_Service	34
Arc::MCC_TLS	35
Arc::MCC_TLS_Client	36
Arc::MCC_TLS_Service	37
Arc::PayloadHTTP	38

Arc::PayloadTCPSocket	43
Arc::PayloadTLSStream	44
ArcSec::PDPServletInvoker	46
ArcSec::SAML2SSO_AssertionConsumerSH	47
ArcSec::SAMLTokenSH	48
ArcSec::SimpleListPDP	49
Arc::SRMClient	50
Arc::SRMClientRequest	59
SRMFileInfo	62
Arc::SRMFileMetaData	63
SRMInfo	64
ArcSec::UsernameTokenSH	65
ArcSec::X509TokenSH	66
ArcSec::XACMLAlgFactory	67
ArcSec::XACMLAttributeFactory	68
ArcSec::XACMLAttributeProxy< TheAttribute >	69
ArcSec::XACMLCondition	70
ArcSec::XACMLEvaluationCtx	71
ArcSec::XACMLEvaluator	72
ArcSec::XACMLFnFactory	73
ArcSec::XACMLPDP	74
ArcSec::XACMLPolicy	75
ArcSec::XACMLRule	76
ArcSec::XACMLTarget	77

Chapter 3

Hosting Environment (Daemon) Chain Components Data Structure Index

3.1 Hosting Environment (Daemon) Chain Components Data Structures

Here are the data structures with brief descriptions:

ArcSec::AllowPDP (This PDP always return true (allow))	11
ArcSec::ArcAlgFactory (Algorithm factory class for Arc)	12
ArcSec::ArcAttributeFactory (Attribute factory class for Arc specified attributes)	13
ArcSec::ArcAttributeProxy < TheAttribute > (Arc specific AttributeProxy class)	14
ArcSec::ArcAuthZ (Tests message against list of PDPs)	15
ArcSec::ArcEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	16
ArcSec::ArcEvaluator (Execute the policy evaluation, based on the request and policy)	17
ArcSec::ArcFnFactory (Function factory class for Arc specified attributes)	18
ArcSec::ArcPDP (ArcPDP (p. 19) - PDP which can handle the Arc specific request and policy schema)	19
ArcSec::ArcPolicy (ArcPolicy (p. 20) class to parse and operate Arc specific <Policy> node)	20
ArcSec::ArcRequestItem (Container, <Subjects, Actions, Objects, Contexts> tuple)	21
ArcSec::ArcRequestTuple (RequestTuple, container which includes the)	22
ArcSec::ArcRule (ArcRule (p. 23) class to parse Arc specific <Rule> node)	23
ArcSec::DelegationPDP	24
ArcSec::DenyPDP (This PDP always returns false (deny))	25
Arc::LDAPQuery	26
Arc::MCC_HTTP (A base class for HTTP client and service MCCs)	27
Arc::MCC_HTTP_Client	28
Arc::MCC_HTTP_Service	29
Arc::MCC_SOAP (A base class for SOAP client and service MCCs)	30
Arc::MCC_SOAP_Service	31
Arc::MCC_TCP (A base class for TCP client and service MCCs)	32
Arc::MCC_TCP_Client	33
Arc::MCC_TCP_Service	34
Arc::MCC_TLS (A base class for TLS client and service MCCs)	35
Arc::MCC_TLS_Client	36
Arc::MCC_TLS_Service	37

Arc::PayloadHTTP	38
Arc::PayloadTCPSocket	43
Arc::PayloadTLSStream	44
ArcSec::PDPServiceInvoker (PDPServiceInvoker (p. 46) - client which will invoke pdpser- vice)	46
ArcSec::SAML2SSO_AssertionConsumerSH (Implement the functionality of the Service Provider in SAML2 SSO profile)	47
ArcSec::SAMLTokenSH (Adds WS-Security SAML Token into SOAP Header)	48
ArcSec::SimpleListPDP (Tests X509 subject against list of subjects in file)	49
Arc::SRMClient	50
Arc::SRMClientRequest	59
SRMFileInfo	62
Arc::SRMFileMetaData	63
SRMInfo	64
ArcSec::UsernameTokenSH (Adds WS-Security Username Token into SOAP Header)	65
ArcSec::X509TokenSH (Adds WS-Security X509 Token into SOAP Header)	66
ArcSec::XACMLAlgFactory (Algorithm factory class for XACML)	67
ArcSec::XACMLAttributeFactory (Attribute factory class for XACML specified attributes)	68
ArcSec::XACMLAttributeProxy < TheAttribute > (XACML specific AttributeProxy class)	69
ArcSec::XACMLCondition (XACMLCondition (p. 70) class to parse and operate XACML specific <Condition> node)	70
ArcSec::XACMLEvaluationCtx (EvaluationCtx, in charge of storing some context infor- mation for evaluation, including Request, current time, etc)	71
ArcSec::XACMLEvaluator (Execute the policy evaluation, based on the request and policy)	72
ArcSec::XACMLFnFactory (Function factory class for XACML specified attributes)	73
ArcSec::XACMLPDP (XACMLPDP (p. 74) - PDP which can handle the XACML specific request and policy schema)	74
ArcSec::XACMLPolicy (XACMLPolicy (p. 75) class to parse and operate XACML specific <Policy> node)	75
ArcSec::XACMLRule (XACMLRule (p. 76) class to parse XACML specific <Rule> node)	76
ArcSec::XACMLTarget (XACMLTarget (p. 77) class to parse and operate XACML specific <Target> node)	77

Chapter 4

Hosting Environment (Daemon) Chain Components Namespace Documentation

4.1 ArcSec Namespace Reference

ArcRequest, Parsing the specified Arc request format.

Data Structures

- class **DelegationCollector**
- class **DelegationSecAttr**
- class **DelegationMultiSecAttr**
- class **AllowPDP**

This PDP always return true (allow).

- class **ArcAuthZ**

Tests message against list of PDPs.

- class **ArcAlgFactory**

Algorithm factory class for Arc.

- class **ArcAttributeFactory**

Attribute factory class for Arc specified attributes.

- class **ArcAttributeProxy**

Arc specific AttributeProxy class.

- class **ArcRequestTuple**

RequestTuple, container which includes the.

- class **ArcEvaluationCtx**

EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.

- **class ArcEvaluator**
Execute the policy evaluation, based on the request and policy.
- **class ArcFnFactory**
Function factory class for Arc specified attributes.
- **class ArcPDP**
ArcPDP (p. 19) - PDP which can handle the Arc specific request and policy schema.
- **class ArcPolicy**
ArcPolicy (p. 20) class to parse and operate Arc specific <Policy> node.
- **class ArcRequest**
- **class ArcRequestItem**
Container, <Subjects, Actions, Objects, Contexts> tuple.
- **class ArcRule**
ArcRule (p. 23) class to parse Arc specific <Rule> node.
- **class DelegationPDP**
- **class DelegationSH**
- **class DenyPDP**
This PDP always returns false (deny).
- **class GACLEvaluator**
- **class GACLPDP**
- **class GACLPolicy**
- **class GACLRequest**
- **class PDPServiceInvoker**
PDPServiceInvoker (p. 46) - client which will invoke pdpservice.
- **class SAML2SSO_AssertionConsumerSH**
Implement the functionality of the Service Provider in SAML2 SSO profile.
- **class SAMLTokenSH**
Adds WS-Security SAML Token into SOAP Header.
- **class SimpleListPDP**
Tests X509 subject against list of subjects in file.
- **class UsernameTokenSH**
Adds WS-Security Username Token into SOAP Header.
- **class X509TokenSH**
Adds WS-Security X509 Token into SOAP Header.
- **class AttributeDesignator**
- **class AttributeSelector**
- **class XACMLAlgFactory**

Algorithm factory class for XACML.

- **class XACMLApply**
- **class XACMLAttributeFactory**
Attribute factory class for XACML specified attributes.
- **class XACMLAttributeProxy**
XACML specific AttributeProxy class.
- **class XACMLCondition**
XACMLCondition (p. 70) class to parse and operate XACML specific <Condition> node.
- **class XACMLEvaluationCtx**
EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.
- **class XACMLEvaluator**
Execute the policy evaluation, based on the request and policy.
- **class XACMLFnFactory**
Function factory class for XACML specified attributes.
- **class XACMLPDP**
XACMLPDP (p. 74) - PDP which can handle the XACML specific request and policy schema.
- **class XACMLPolicy**
XACMLPolicy (p. 75) class to parse and operate XACML specific <Policy> node.
- **class XACMLRequest**
- **class XACMLRule**
XACMLRule (p. 76) class to parse XACML specific <Rule> node.
- **class XACMLTargetMatch**
- **class XACMLTargetMatchGroup**
- **class XACMLTargetSection**
- **class XACMLTarget**
XACMLTarget (p. 77) class to parse and operate XACML specific <Target> node.

Typedefs

- **typedef std::pair< AttributeValue *, Function * > Match**
- **typedef std::list< Match > AndList**
- **typedef std::list< AndList > OrList**

4.1.1 Detailed Description

ArcRequest, Parsing the specified Arc request format.

4.1.2 Typedef Documentation

4.1.2.1 typedef std::pair<AttributeValue*, Function*> ArcSec::Match

Pair Match include the AttributeValue object in <Rule> and the Function which is used to handle the AttributeValue, default function is "Equal", if some other function is used, it should be explicitly specified, e.g. Subject Type="string" Function="Match"/>/vo.knowarc/usergroup-A</Subject>Subjects> example inside <Rule>: <Subjects> <Subject type="X500Name">/O=Nordu-Grid/OU=UIO/CN=test</Subject> <Subject type="string">/vo.knowarc/usergroupA</Subject> <Subject> <SubFraction type="string">/O=Grid/OU=KnowARC/CN=XYZ</SubFraction> <SubFraction type="string">urn:mace:shibboleth:examples</SubFraction> </Subject> <GroupIdRef location="/.subjectgroup.xml">subgrpexample1</GroupIdRef> </Subjects>

4.1.2.2 typedef std::list<Match> ArcSec::AndList

AndList - include items inside one <Subject> (or <Resource> <Action> <Condition>).

"Or" relationship meand the request should satisfy any of the items <Subjects>
 <Subject type="X500DN">/O=Grid/OU=KnowARC/CN=ABC</Subject> <Subject
 type="VOMSAttribute">/vo.knowarc/usergroupA</Subject> <Subject> <SubFraction
 type="X500DN">/O=Grid/OU=KnowARC/CN=XYZ</SubFraction> <SubFraction
 type="ShibName">urn:mace:shibboleth:examples</SubFraction> </Subject> <GroupIdRef
 location="/.subjectgroup.xml">subgrpexample1</GroupIdRef> </Subjects>

4.1.2.3 typedef std::list<AndList> ArcSec::OrList

OrList - include items inside one <Subjects> (or <Resources> <Actions> <Conditions>).

Chapter 5

Hosting Environment (Daemon) Chain Components Data Structure Documentation

5.1 ArcSec::AllowPDP Class Reference

This PDP always return true (allow).

```
#include <AllowPDP.h>
```

5.1.1 Detailed Description

This PDP always return true (allow).

The documentation for this class was generated from the following file:

- AllowPDP.h

5.2 ArcSec::ArcAlgFactory Class Reference

Algorithm factory class for Arc.

```
#include <ArcAlgFactory.h>
```

Public Member Functions

- virtual CombiningAlg * **createAlg** (const std::string &type)

5.2.1 Detailed Description

Algorithm factory class for Arc.

5.2.2 Member Function Documentation

5.2.2.1 virtual CombiningAlg* ArcSec::ArcAlgFactory::createAlg (const std::string & type) [virtual]

return a Alg object according to the "CombiningAlg" attribute in the <Policy> node; The **ArcAlgFactory** (p. 12) itself will release the Alg objects

The documentation for this class was generated from the following file:

- ArcAlgFactory.h

5.3 ArcSec::ArcAttributeFactory Class Reference

Attribute factory class for Arc specified attributes.

```
#include <ArcAttributeFactory.h>
```

Public Member Functions

- virtual AttributeValue * **createValue** (const Arc::XMLNode &node, const std::string &type)

5.3.1 Detailed Description

Attribute factory class for Arc specified attributes.

5.3.2 Member Function Documentation

5.3.2.1 virtual AttributeValue* ArcSec::ArcAttributeFactory::createValue (const Arc::XMLNode & node, const std::string & type) [virtual]

creat a AttributeValue according to the value in the XML node and the type; It should be the caller to release the AttributeValue Object

The documentation for this class was generated from the following file:

- ArcAttributeFactory.h

5.4 ArcSec::ArcAttributeProxy< TheAttribute > Class Template Reference

Arc specific AttributeProxy class.

```
#include <ArcAttributeProxy.h>
```

Public Member Functions

- virtual AttributeValue * **getAttribute** (const Arc::XMLNode &node)

5.4.1 Detailed Description

```
template<class TheAttribute> class ArcSec::ArcAttributeProxy< TheAttribute >
```

Arc specific AttributeProxy class.

5.4.2 Member Function Documentation

5.4.2.1 `template<class TheAttribute> AttributeValue * ArcSec::ArcAttributeProxy< TheAttribute >::getAttribute (const Arc::XMLNode & node) [virtual]`

Implementation of getAttribute method.

The documentation for this class was generated from the following file:

- ArcAttributeProxy.h

5.5 ArcSec::ArcAuthZ Class Reference

Tests message against list of PDPs.

```
#include <ArcAuthZ.h>
```

Public Member Functions

- virtual bool **Handle** (Arc::Message *msg) const

Protected Member Functions

- bool **MakePDPs** (Arc::XMLNode cfg)

Data Structures

- class **PDPDesc**

5.5.1 Detailed Description

Tests message against list of PDPs.

This class implements SecHandler interface. It's **Handle**() (p. 15) method runs provided Message instance against all PDPs specified in configuration. If any of PDPs returns positive result **Handle**() (p. 15) return true, otherwise false. This class is the main entry for configuring authorization, and could include different PDP configured inside.

5.5.2 Member Function Documentation

5.5.2.1 virtual bool ArcSec::ArcAuthZ::Handle (Arc::Message * msg) const [virtual]

Get authorization decision

5.5.2.2 bool ArcSec::ArcAuthZ::MakePDPs (Arc::XMLNode cfg) [protected]

Create PDP according to conf info

The documentation for this class was generated from the following file:

- ArcAuthZ.h

5.6 ArcSec::ArcEvaluationCtx Class Reference

EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.

```
#include <ArcEvaluationCtx.h>
```

Public Member Functions

- **ArcEvaluationCtx (Request *request)**
- **virtual void split ()**

5.6.1 Detailed Description

EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 ArcSec::ArcEvaluationCtx::ArcEvaluationCtx (Request * *request*)

Construct a new EvaluationCtx based on the given request

5.6.3 Member Function Documentation

5.6.3.1 virtual void ArcSec::ArcEvaluationCtx::split () [virtual]

Convert/split one RequestItem (one tuple <SubList, ResList, ActList, CtxList>) into a few <Subject, Resource, Action, Context> tuples. The purpose is for evaluation. The evaluator will evaluate each Request-Tuple one by one, not the RequestItem because it includes some independent <Subject, Resource, Action, Context>s and the evaluator should deal with them independently.

The documentation for this class was generated from the following file:

- ArcEvaluationCtx.h

5.7 ArcSec::ArcEvaluator Class Reference

Execute the policy evaluation, based on the request and policy.

```
#include <ArcEvaluator.h>
```

Public Member Functions

- virtual Response * **evaluate** (Request *request)

5.7.1 Detailed Description

Execute the policy evaluation, based on the request and policy.

5.7.2 Member Function Documentation

5.7.2.1 virtual Response* ArcSec::ArcEvaluator::evaluate (Request *request) [virtual]

Evaluate the request based on the policy information inside PolicyStore

The documentation for this class was generated from the following file:

- ArcEvaluator.h

5.8 ArcSec::ArcFnFactory Class Reference

Function factory class for Arc specified attributes.

```
#include <ArcFnFactory.h>
```

Public Member Functions

- virtual Function * **createFn** (const std::string &type)

5.8.1 Detailed Description

Function factory class for Arc specified attributes.

5.8.2 Member Function Documentation

5.8.2.1 virtual Function* ArcSec::ArcFnFactory::createFn (const std::string & type) [virtual]

return a Function object according to the "Function" attribute in the XML node; The **ArcFnFactory** (p. 18) itself will release the Function objects

The documentation for this class was generated from the following file:

- ArcFnFactory.h

5.9 ArcSec::ArcPDP Class Reference

ArcPDP (p. 19) - PDP which can handle the Arc specific request and policy schema.

```
#include <ArcPDP.h>
```

5.9.1 Detailed Description

ArcPDP (p. 19) - PDP which can handle the Arc specific request and policy schema.

The documentation for this class was generated from the following file:

- ArcPDP.h

5.10 ArcSec::ArcPolicy Class Reference

ArcPolicy (p. 20) class to parse and operate Arc specific <Policy> node.

```
#include <ArcPolicy.h>
```

Public Member Functions

- **ArcPolicy** (void)
- **ArcPolicy** (const Arc::XMLNode node)
- **ArcPolicy** (const Arc::XMLNode node, EvaluatorContext *ctx)
- **virtual void make_policy** ()

5.10.1 Detailed Description

ArcPolicy (p. 20) class to parse and operate Arc specific <Policy> node.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 ArcSec::ArcPolicy::ArcPolicy (void)

Constructor

5.10.2.2 ArcSec::ArcPolicy::ArcPolicy (const Arc::XMLNode *node*)

Constructor

5.10.2.3 ArcSec::ArcPolicy::ArcPolicy (const Arc::XMLNode *node*, EvaluatorContext * *ctx*)

Constructor

5.10.3 Member Function Documentation

5.10.3.1 **virtual void ArcSec::ArcPolicy::make_policy** () [virtual]

Parse XMLNode, and construct the low-level Rule object

The documentation for this class was generated from the following file:

- ArcPolicy.h

5.11 ArcSec::ArcRequestItem Class Reference

Container, <Subjects, Actions, Objects, Contexts> tuple.

```
#include <ArcRequestItem.h>
```

5.11.1 Detailed Description

Container, <Subjects, Actions, Objects, Contexts> tuple.

Specified **ArcRequestItem** (p. 21) which can parse Arc request formate

The documentation for this class was generated from the following file:

- ArcRequestItem.h

5.12 ArcSec::ArcRequestTuple Class Reference

RequestTuple, container which includes the.

```
#include <ArcEvaluationCtx.h>
```

5.12.1 Detailed Description

RequestTuple, container which includes the.

The documentation for this class was generated from the following file:

- ArcEvaluationCtx.h

5.13 ArcSec::ArcRule Class Reference

ArcRule (p. 23) class to parse Arc specific <Rule> node.

```
#include <ArcRule.h>
```

5.13.1 Detailed Description

ArcRule (p. 23) class to parse Arc specific <Rule> node.

The documentation for this class was generated from the following file:

- ArcRule.h

5.14 ArcSec::DelegationPDP Class Reference

```
#include <DelegationPDP.h>
```

5.14.1 Detailed Description

DeleagtionPDP - PDP which can handle the Arc specific request and policy provided as identity delegation policy.

The documentation for this class was generated from the following file:

- DelegationPDP.h

5.15 ArcSec::DenyPDP Class Reference

This PDP always returns false (deny).

```
#include <DenyPDP.h>
```

5.15.1 Detailed Description

This PDP always returns false (deny).

The documentation for this class was generated from the following file:

- DenyPDP.h

5.16 Arc::LDAPQuery Class Reference

```
#include <LDAPQuery.h>
```

Public Member Functions

- **LDAPQuery** (const std::string &ldaphost, int ldapport, int timeout, bool anonymous=true, const std::string &usersn="")
- **~LDAPQuery** ()
- **bool Query** (const std::string &base, const std::string &filter="(objectclass=*)", const std::list< std::string > &attributes=std::list< std::string >(), URL::Scope scope=URL::subtree)
- **bool Result** (ldap_callback callback, void *ref)

5.16.1 Detailed Description

LDAPQuery (p. 26) class; querying of LDAP servers.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Arc::LDAPQuery::LDAPQuery (const std::string &ldaphost, int ldapport, int timeout, bool anonymous = true, const std::string &usersn = "")

Constructs a new **LDAPQuery** (p. 26) object and sets connection options. The connection is first established when calling **Query**.

5.16.2.2 Arc::LDAPQuery::~~LDAPQuery ()

Destructor. Will disconnect from the ldapservers if still connected.

5.16.3 Member Function Documentation

5.16.3.1 bool Arc::LDAPQuery::Query (const std::string &base, const std::string &filter = "(objectclass=*)", const std::list< std::string > &attributes = std::list< std::string >(), URL::Scope scope = URL::subtree)

Queries the ldap server.

5.16.3.2 bool Arc::LDAPQuery::Result (ldap_callback callback, void *ref)

Retrieves the result of the query from the ldap-server.

The documentation for this class was generated from the following file:

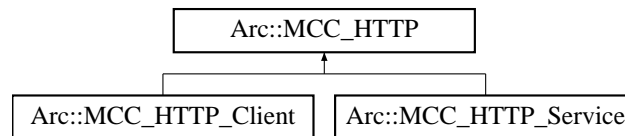
- LDAPQuery.h

5.17 Arc::MCC_HTTP Class Reference

A base class for HTTP client and service MCCs.

```
#include <MCCHTTP.h>
```

Inheritance diagram for Arc::MCC_HTTP::



5.17.1 Detailed Description

A base class for HTTP client and service MCCs.

This is a base class for HTTP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

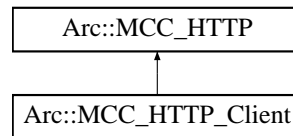
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.18 Arc::MCC_HTTP_Client Class Reference

```
#include <MCCHTTP.h>
```

Inheritance diagram for Arc::MCC_HTTP_Client::



5.18.1 Detailed Description

This class is a client part of HTTP MCC. It accepts PayloadRawInterface payload and uses it as body to generate HTTP request. Request is passed to next MCC as PayloadRawInterface type of payload. Returned PayloadStreamInterface payload is parsed into HTTP response and it's body is passed back to calling MCC as PayloadRawInterface. Attributes of request/input message of type HTTP:name are translated into HTTP header with corresponding 'name's. Special attributes HTTP:METHOD and HTTP:ENDPOINT specify method and URL in HTTP request. If not present meathod and URL are taken from configuration. In output/response message following attributes are present: HTTP:CODE - response code of HTTP HTTP:REASON - reason string of HTTP response HTTP:name - all 'name' attributes of HTTP header.

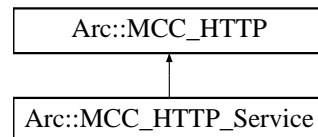
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.19 Arc::MCC_HTTP_Service Class Reference

```
#include <MCCHTTP.h>
```

Inheritance diagram for Arc::MCC_HTTP_Service::



5.19.1 Detailed Description

This class implements MCC to processes HTTP request. On input payload with PayloadStreamInterface is expected. HTTP message is read from stream and its body is converted into PayloadRaw and passed to next MCC. Returned payload of PayloadRawInterface type is treated as body part of returning **Payload-HTTP** (p. 38). Generated HTTP response is sent through stream passed in input payload. During processing of request/input message following attributes are generated: HTTP:METHOD - HTTP method e.g. GET, PUT, POST, etc. HTTP:ENDPOINT - URL taken from HTTP request ENDPOINT - global attribute equal to HTTP:ENDPOINT HTTP:RANGESTART - start of requested byte range HTTP:RANGEEND - end of requested byte range (inclusive) HTTP:name - all 'name' attributes of HTTP header. Attributes of response message of HTTP:name type are translated into HTTP header with corresponding 'name's.

The documentation for this class was generated from the following file:

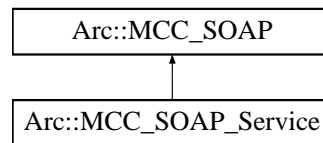
- MCCHTTP.h

5.20 Arc::MCC_SOAP Class Reference

A base class for SOAP client and service MCCs.

```
#include <MCCSOAP.h>
```

Inheritance diagram for Arc::MCC_SOAP::



5.20.1 Detailed Description

A base class for SOAP client and service MCCs.

This is a base class for SOAP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

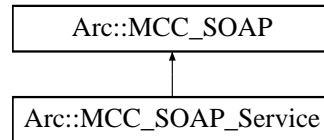
The documentation for this class was generated from the following file:

- MCCSOAP.h

5.21 Arc::MCC_SOAP_Service Class Reference

```
#include <MCCSOAP.h>
```

Inheritance diagram for Arc::MCC_SOAP_Service::



5.21.1 Detailed Description

This MCC parses SOAP message from input payload. On input payload with PayloadRawInterface is expected. It's converted into PayloadSOAP and passed next MCC. Returned PayloadSOAP is converted into PayloadRaw and returned to calling MCC.

The documentation for this class was generated from the following file:

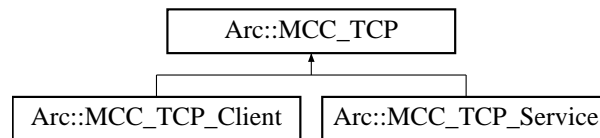
- MCCSOAP.h

5.22 Arc::MCC_TCP Class Reference

A base class for TCP client and service MCCs.

```
#include <MCCTCP.h>
```

Inheritance diagram for Arc::MCC_TCP::



5.22.1 Detailed Description

A base class for TCP client and service MCCs.

This is a base class for TCP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

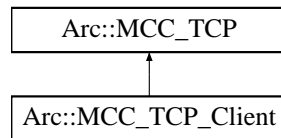
The documentation for this class was generated from the following file:

- MCCTCP.h

5.23 Arc::MCC_TCP_Client Class Reference

```
#include <MCCTCP.h>
```

Inheritance diagram for Arc::MCC_TCP_Client::



5.23.1 Detailed Description

This class is MCC implementing TCP client. Upon creation it connects to specified TCP post at specified host. process() method accepts PayloadRawInterface type of payload. Content of payload is sent over TCP socket. It returns PayloadStreamInterface payload for previous MCC to read response.

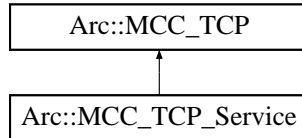
The documentation for this class was generated from the following file:

- MCCTCP.h

5.24 Arc::MCC_TCP_Service Class Reference

```
#include <MCCTCP.h>
```

Inheritance diagram for Arc::MCC_TCP_Service::



Public Member Functions

- **MCC_TCP_Service (Config *cfg)**

Data Structures

- class **mcc_tcp_exec_t**
- class **mcc_tcp_handle_t**

5.24.1 Detailed Description

This class is MCC implementing TCP server. Upon creation this object binds to specified TCP ports and listens for incoming TCP connections on dedicated thread. Each connection is accepted and dedicated thread is created. Then that thread is used to call process() method of next MCC in chain. That method is passed payload implementing PayloadStreamInterface. On response payload with PayloadRawInterface is expected. Alternatively called MCC may use provided PayloadStreamInterface to send it's response back directly. During processing of request this MCC generates following attributes: TCP:HOST - IP address of interface to which local TCP socket is bound TCP:PORT - port number to which local TCP socket is bound TCP:REMOTEHOST - IP address from which connection is accepted TCP:REMOTEPORT - TCP port from which connection is accepted TCP:ENDPOINT - URL-like representation of remote connection - ://HOST:PORT ENDPOINT - global attribute equal to TCP:ENDPOINT

5.24.2 Constructor & Destructor Documentation

5.24.2.1 Arc::MCC_TCP_Service::MCC_TCP_Service (Config *cfg)

executing function for connection thread

The documentation for this class was generated from the following file:

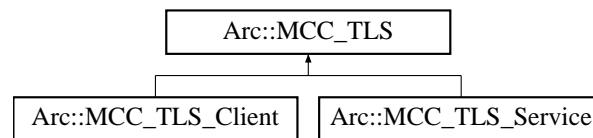
- MCCTCP.h

5.25 Arc::MCC_TLS Class Reference

A base class for TLS client and service MCCs.

```
#include <MCCTLS.h>
```

Inheritance diagram for Arc::MCC_TLS::



5.25.1 Detailed Description

A base class for TLS client and service MCCs.

This is a base class for TLS client and service MCCs. It provides some common functionality for them.

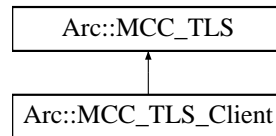
The documentation for this class was generated from the following file:

- MCCTLS.h

5.26 Arc::MCC_TLS_Client Class Reference

```
#include <MCCTLS.h>
```

Inheritance diagram for Arc::MCC_TLS_Client::



5.26.1 Detailed Description

This class is MCC implementing TLS client.

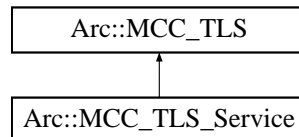
The documentation for this class was generated from the following file:

- MCCTLS.h

5.27 Arc::MCC_TLS_Service Class Reference

```
#include <MCCTLS.h>
```

Inheritance diagram for Arc::MCC_TLS_Service::



5.27.1 Detailed Description

This MCC implements TLS server side functionality. Upon creation this object creates SSL_CTX object and configures SSL_CTX object with some environment information about credential. Because we cannot know the "socket" when the creation of MCC_TLS_Service/MCC_TLS_Client object (not like **MCC_TCP_Client** (p. 33), which can creat socket in the constructor method by using information in configuration file), we can only creat "ssl" object which is binded to specified "socket", when **MCC_HTTP_Client** (p. 28) calls the process() method of **MCC_TLS_Client** (p. 36) object, or **MCC_TCP_Service** (p. 34) calls the process() method of **MCC_TLS_Service** (p. 37) object. The "ssl" object is embeded in a payload called PayloadTLSSocket.

The process() method of **MCC_TLS_Service** (p. 37) is passed payload implementing PayloadStream-Interface and the method returns empty PayloadRaw payload in "outmsg". The ssl object is created and bound to Stream payload when constructing the PayloadTLSSocket in the process() method.

During processing of message this MCC generates attribute TLS:PEERDN which contains Distinguished Name of remoote peer.

The documentation for this class was generated from the following file:

- MCCTLS.h

5.28 Arc::PayloadHTTP Class Reference

```
#include <PayloadHTTP.h>
```

Public Member Functions

- **PayloadHTTP** (PayloadStreamInterface &stream, bool own=false)
- **PayloadHTTP** (const std::string &method, const std::string &url, PayloadStreamInterface &stream)
- **PayloadHTTP** (const std::string &method, const std::string &url)
- **PayloadHTTP** (int code, const std::string &reason, PayloadStreamInterface &stream)
- **PayloadHTTP** (int code, const std::string &reason)
- **virtual const std::string &Attribute** (const std::string &name)
- **virtual const std::multimap< std::string, std::string > &Attributes** (void)
- **virtual void Attribute** (const std::string &name, const std::string &value)
- **virtual bool Flush** (void)
- **virtual void Body** (PayloadRawInterface &body, bool ownership=true)

Protected Member Functions

- **bool readline** (std::string &line)
- **bool read** (char *buf, int64_t &size)
- **bool parse_header** (void)
- **bool get_body** (void)

Protected Attributes

- PayloadStreamInterface * **stream_**
- bool **stream_own_**
- PayloadRawInterface * **rbody_**
- PayloadStreamInterface * **sbody_**
- bool **body_own_**
- std::string **uri_**
- int **version_major_**
- int **version_minor_**
- std::string **method_**
- int **code_**
- std::string **reason_**
- int64_t **length_**
- bool **chunked_**
- bool **keep_alive_**
- std::multimap< std::string, std::string > **attributes_**

5.28.1 Detailed Description

This class implements parsing and generation of HTTP messages. It implements only subset of HTTP/1.1 and also provides an PayloadRawInterface for including as payload into Message passed through MCC chains.

5.28.2 Constructor & Destructor Documentation

5.28.2.1 Arc::PayloadHTTP::PayloadHTTP (PayloadStreamInterface & *stream*, bool *own* = false)

Constructor - creates object by parsing HTTP request or response from stream. Supplied stream is associated with object for later use. If *own* is set to true then stream will be deleted in destructor. Because stream can be used by this object during whole lifetime it is important not to destroy stream till this object is deleted.

5.28.2.2 Arc::PayloadHTTP::PayloadHTTP (const std::string & *method*, const std::string & *url*, PayloadStreamInterface & *stream*)

Constructor - creates HTTP request to be sent through stream. HTTP message is not sent yet.

5.28.2.3 Arc::PayloadHTTP::PayloadHTTP (const std::string & *method*, const std::string & *url*)

Constructor - creates HTTP request to be rendered through Raw interface.

5.28.2.4 Arc::PayloadHTTP::PayloadHTTP (int *code*, const std::string & *reason*, PayloadStreamInterface & *stream*)

Constructor - creates HTTP response to be sent through stream. HTTP message is not sent yet.

5.28.2.5 Arc::PayloadHTTP::PayloadHTTP (int *code*, const std::string & *reason*)

Constructor - creates HTTP response to be rendered through Raw interface.

5.28.3 Member Function Documentation

5.28.3.1 virtual void Arc::PayloadHTTP::Attribute (const std::string & *name*, const std::string & *value*) [virtual]

Adds HTTP header attribute 'name' = 'value'

5.28.3.2 virtual const std::string& Arc::PayloadHTTP::Attribute (const std::string & *name*) [virtual]

Returns HTTP header attribute with specified name. Empty string if no such attribute.

5.28.3.3 virtual const std::multimap<std::string, std::string>& Arc::PayloadHTTP::Attributes (void) [virtual]

Returns all HTTP header attributes.

5.28.3.4 `virtual void Arc::PayloadHTTP::Body (PayloadRawInterface & body, bool ownership = true) [virtual]`

Assign HTTP body. Assigned object is not copied. Instead it is remembered and made available through Raw interface. If 'ownership' is true then passed object is treated as being owned by this instance and destroyed in destructor.

5.28.3.5 `virtual bool Arc::PayloadHTTP::Flush (void) [virtual]`

Send created object through associated stream. If there is no stream associated then HTTP specific data is inserted into Raw buffers of this object. In last case this operation should not be repeated till content of buffer is completely rewritten.

5.28.3.6 `bool Arc::PayloadHTTP::get_body (void) [protected]`

Read Body of HTTP message and attach it to inherited PayloadRaw object

5.28.3.7 `bool Arc::PayloadHTTP::parse_header (void) [protected]`

Read HTTP header and fill internal variables

5.28.3.8 `bool Arc::PayloadHTTP::read (char * buf, int64_t & size) [protected]`

Read up to 'size' bytes from stream_

5.28.3.9 `bool Arc::PayloadHTTP::readline (std::string & line) [protected]`

Read from stream till

5.28.4 Field Documentation

5.28.4.1 `std::multimap<std::string, std::string> Arc::PayloadHTTP::attributes_ [protected]`

true if connection should not be closed after response

5.28.4.2 `bool Arc::PayloadHTTP::body_own_ [protected]`

associated HTTP Body stream if any (to avoid copying to own buffer)

5.28.4.3 `bool Arc::PayloadHTTP::chunked_ [protected]`

Content-length of HTTP message

5.28.4.4 `int Arc::PayloadHTTP::code_ [protected]`

HTTP method being used or requested

5.28.4.5 `bool Arc::PayloadHTTP::keep_alive_` [protected]

true if content is chunked

5.28.4.6 `int64_t Arc::PayloadHTTP::length_` [protected]

HTTP reason being sent or supplied

5.28.4.7 `std::string Arc::PayloadHTTP::method_` [protected]

minor number of HTTP version - must be 0 or 1

5.28.4.8 `PayloadRawInterface* Arc::PayloadHTTP::rbody_` [protected]

if true stream_ is owned by this

5.28.4.9 `std::string Arc::PayloadHTTP::reason_` [protected]

HTTP code being sent or supplied

5.28.4.10 `PayloadStreamInterface* Arc::PayloadHTTP::sbody_` [protected]

associated HTTP Body buffer if any (to avoid copying to own buffer)

5.28.4.11 `PayloadStreamInterface* Arc::PayloadHTTP::stream_` [protected]

true if whole content of HTTP body was fetched and stored in buffers. Otherwise only header was fetched and part of body in tbuf_ and rest is to be read through stream_.

5.28.4.12 `bool Arc::PayloadHTTP::stream_own_` [protected]

stream used to communicate to outside

5.28.4.13 `std::string Arc::PayloadHTTP::uri_` [protected]

if true body_ is owned by this

5.28.4.14 `int Arc::PayloadHTTP::version_major_` [protected]

URI being contacted

5.28.4.15 `int Arc::PayloadHTTP::version_minor_` [protected]

major number of HTTP version - must be 1

The documentation for this class was generated from the following file:

- PayloadHTTP.h

5.29 Arc::PayloadTCPSocket Class Reference

```
#include <PayloadTCPSocket.h>
```

Public Member Functions

- **PayloadTCPSocket** (const char *hostname, int port, int timeout, Logger &logger)
- **PayloadTCPSocket** (const std::string endpoint, int timeout, Logger &logger)
- **PayloadTCPSocket** (int s, int timeout, Logger &logger)
- **PayloadTCPSocket** (PayloadTCPSocket &s)
- **PayloadTCPSocket** (PayloadTCPSocket &s, Logger &logger)

5.29.1 Detailed Description

This class extends PayloadStream with TCP socket specific features

5.29.2 Constructor & Destructor Documentation

5.29.2.1 Arc::PayloadTCPSocket::PayloadTCPSocket (const char * *hostname*, int *port*, int *timeout*, Logger & *logger*)

Constructor - connects to TCP server at specified hostname:port

5.29.2.2 Arc::PayloadTCPSocket::PayloadTCPSocket (const std::string *endpoint*, int *timeout*, Logger & *logger*)

Constructor - connects to TCP server at specified endpoint - hostname:port

5.29.2.3 Arc::PayloadTCPSocket::PayloadTCPSocket (int *s*, int *timeout*, Logger & *logger*) [inline]

Constructor - creates object of already connected socket. Socket is NOT closed in destructor.

5.29.2.4 Arc::PayloadTCPSocket::PayloadTCPSocket (PayloadTCPSocket & *s*) [inline]

Copy constructor - inherits socket of copied object. Socket is NOT closed in destructor.

5.29.2.5 Arc::PayloadTCPSocket::PayloadTCPSocket (PayloadTCPSocket & *s*, Logger & *logger*) [inline]

Copy constructor - inherits handle of copied object. Handle is NOT closed in destructor.

The documentation for this class was generated from the following file:

- PayloadTCPSocket.h

5.30 Arc::PayloadTLSStream Class Reference

```
#include <PayloadTLSStream.h>
```

Public Member Functions

- **PayloadTLSStream (Logger &logger, SSL *ssl=NULL)**
- **virtual ~PayloadTLSStream (void)**
- **X509 * GetPeerCert (void)**
- **STACK_OF (X509)*GetPeerChain(void)**
- **X509 * GetCert (void)**

Protected Attributes

- **SSL * ssl_**

5.30.1 Detailed Description

Implemetation of PayloadStreamInterface for SSL handle.

5.30.2 Constructor & Destructor Documentation

5.30.2.1 Arc::PayloadTLSStream::PayloadTLSStream (Logger & *logger*, SSL * *ssl* = NULL)

Constructor. Attaches to already open handle. Handle is not managed by this class and must be closed by external code.

5.30.2.2 virtual Arc::PayloadTLSStream::~~PayloadTLSStream (void) [virtual]

Destructor.

5.30.3 Member Function Documentation

5.30.3.1 X509* Arc::PayloadTLSStream::GetCert (void)

Get local certificate from associated ssl. Obtained X509 object is owned by this instance and becomes invalid after destruction.

5.30.3.2 X509* Arc::PayloadTLSStream::GetPeerCert (void)

Get peer certificate from the established ssl. Obtained X509 object is owned by this instance and becomes invalid after destruction. Still obtained has to be freed at end of usage.

5.30.3.3 Arc::PayloadTLSStream::STACK_OF (X509)

Get chain of peer certificates from the established ssl. Obtained X509 object is owned by this instance and becomes invalid after destruction.

5.30.4 Field Documentation

5.30.4.1 SSL* Arc::PayloadTLSStream::ssl_ [protected]

Timeout for read/write operations

The documentation for this class was generated from the following file:

- PayloadTLSStream.h

5.31 ArcSec::PDPSERVICEInvoker Class Reference

PDPSERVICEInvoker (p. 46) - client which will invoke pdpservice.

```
#include <PDPSERVICEInvoker.h>
```

5.31.1 Detailed Description

PDPSERVICEInvoker (p. 46) - client which will invoke pdpservice.

The documentation for this class was generated from the following file:

- PDPSERVICEInvoker.h

5.32 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference

Implement the functionality of the Service Provider in SAML2 SSO profile.

```
#include <SAML2SSO_AssertionConsumerSH.h>
```

5.32.1 Detailed Description

Implement the functionality of the Service Provider in SAML2 SSO profile.

The documentation for this class was generated from the following file:

- SAML2SSO_AssertionConsumerSH.h

5.33 ArcSec::SAMLTokenSH Class Reference

Adds WS-Security SAML Token into SOAP Header.

```
#include <SAMLTokenSH.h>
```

5.33.1 Detailed Description

Adds WS-Security SAML Token into SOAP Header.

The documentation for this class was generated from the following file:

- SAMLTokenSH.h

5.34 ArcSec::SimpleListPDP Class Reference

Tests X509 subject against list of subjects in file.

```
#include <SimpleListPDP.h>
```

5.34.1 Detailed Description

Tests X509 subject against list of subjects in file.

This class implements PDP interface. It's isPermitted() method compares X509 subject of requestor obtained from TLS layer (TLS:PEERDN) to list of subjects (one per line) in external file. Location of file is defined by 'location' attribute of PDP configuration. Returns true if subject is present in list, otherwise false.

The documentation for this class was generated from the following file:

- SimpleListPDP.h

5.35 Arc::SRMClient Class Reference

```
#include <SRMClient.h>
```

Public Member Functions

- virtual ~SRMClient ()
- std::string getVersion () const
- virtual SRMReturnCode ping (std::string &version, bool report_error=true)=0
- virtual SRMReturnCode getSpaceTokens (std::list< std::string > &tokens, const std::string &description="")=0
- virtual SRMReturnCode getRequestTokens (std::list< std::string > &tokens, const std::string &description="")=0
- virtual SRMReturnCode getTURLs (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode getTURLsStatus (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode requestBringOnline (SRMClientRequest &req)=0
- virtual SRMReturnCode requestBringOnlineStatus (SRMClientRequest &req)=0
- virtual SRMReturnCode putTURLs (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode putTURLsStatus (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode releaseGet (SRMClientRequest &req)=0
- virtual SRMReturnCode releasePut (SRMClientRequest &req)=0
- virtual SRMReturnCode release (SRMClientRequest &req)=0
- virtual SRMReturnCode abort (SRMClientRequest &req)=0
- virtual SRMReturnCode info (SRMClientRequest &req, std::list< struct SRMFileMetaData > &metadata, const int recursive=0, bool report_error=true)=0
- virtual SRMReturnCode remove (SRMClientRequest &req)=0
- virtual SRMReturnCode copy (SRMClientRequest &req, const std::string &source)=0
- virtual SRMReturnCode mkdir (SRMClientRequest &req)=0

Static Public Member Functions

- static SRMClient * getInstance (const UserConfig &usercfg, const std::string &url, bool &timedout, time_t conn_timeout=60)

Protected Member Functions

- SRMClient (const UserConfig &usercfg, const SRMURL &url)
- SRMReturnCode process (PayloadSOAP *request, PayloadSOAP **response)

Protected Attributes

- std::string service_endpoint
- MCCCConfig cfg
- ClientSOAP * client
- NS ns

- SRMImplementation implementation
- time_t user_timeout
- std::string version

Static Protected Attributes

- static time_t connection_timeout
- static Logger logger

5.35.1 Detailed Description

A client interface to the SRM protocol. Instances of SRM clients are created by calling the **getInstance()** (p. 52) factory method. One client instance can be used to make many requests to the same server (with the same protocol version), but not multiple servers.

5.35.2 Constructor & Destructor Documentation

5.35.2.1 Arc::SRMClient::SRMClient (const UserConfig & *usercfg*, const SRMURL & *url*) [protected]

Constructor

5.35.2.2 virtual Arc::SRMClient::~~SRMClient () [virtual]

Destructor

5.35.3 Member Function Documentation

5.35.3.1 virtual SRMReturnCode Arc::SRMClient::abort (SRMClientRequest & *req*) [pure virtual]

Called in the case of failure during transfer or releasePut. Releases all TURLs involved in the transfer.

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.2 virtual SRMReturnCode Arc::SRMClient::copy (SRMClientRequest & *req*, const std::string & *source*) [pure virtual]

Copy a file between two SRM storages.

Parameters:

req The request object

source The source SURL

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.3 static SRMClient* Arc::SRMClient::getInstance (const UserConfig & *usercfg*, const std::string & *url*, bool & *timedout*, time_t *conn_timeout* = 60) [static]

Returns an **SRMClient** (p.50) instance with the required protocol version. This must be used to create **SRMClient** (p.50) instances. Specifying a version explicitly forces creation of a client with that version.

Parameters:

usercfg The user configuration.

url A SURL. A client connects to the service host derived from this SURL. All operations with a client instance must use SURLs with the same host as this one.

timedout Whether the connection timed out

conn_timeout Connection timeout to the SRM service

Returns:

A pointer to an instance of **SRMClient** (p.50) is returned, or NULL if it was not possible to create one.

5.35.3.4 virtual SRMReturnCode Arc::SRMClient::getRequestTokens (std::list< std::string > & *tokens*, const std::string & *description* = "") [pure virtual]

Returns a list of request tokens for the user calling the method which are still active requests, or the tokens corresponding to the token description, if given.

Parameters:

tokens The list filled by the service

description The user request description, which can be specified when the request is created

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.5 virtual SRMReturnCode Arc::SRMClient::getSpaceTokens (std::list< std::string > & *tokens*, const std::string & *description* = "") [pure virtual]

Find the space tokens available to write to which correspond to the space token description, if given. The list of tokens is a list of numbers referring to the SRM internal definition of the spaces, not user-readable strings.

Parameters:

tokens The list filled by the service

description The space token description

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.6 virtual SRMReturnCode Arc::SRMClient::getTURLs (SRMClientRequest & req, std::list< std::string > & urls) [pure virtual]

If the user wishes to copy a file from somewhere, **getTURLs()** (p. 53) is called to retrieve the transport URL(s) to copy the file from. It may be used synchronously or asynchronously, depending on the synchronous property of the request object. In the former case it will block until the TURLs are ready, in the latter case it will return after making the request and **getTURLsStatus()** (p. 53) must be used to poll the request status if it was not completed.

Parameters:

req The request object

urls A list of TURLs filled by the method

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.7 virtual SRMReturnCode Arc::SRMClient::getTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [pure virtual]

In the case where **getTURLs** was called asynchronously and the request was not completed, this method should be called to poll the status of the request. **getTURLs** must be called before this method and the request object must have ongoing request status.

Parameters:

req The request object. Status must be ongoing.

urls A list of TURLs filled by the method if the request completed successfully

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.8 std::string Arc::SRMClient::getVersion () const [inline]

Returns the version of the SRM protocol used by this instance

5.35.3.9 virtual SRMReturnCode Arc::SRMClient::info (SRMClientRequest & req, std::list< struct SRMFileMetaData > & metadata, const int recursive = 0, bool report_error = true) [pure virtual]

Returns information on a file or files (v2.2 and higher) stored in an SRM, such as file size, checksum and estimated access latency.

Parameters:

req The request object
metadata A list of structs filled with file information
recursive The level of recursion into sub directories
report_error Determines if errors should be reported

Returns:

SRMReturnCode specifying outcome of operation

See also:

SRMFileMetaData (p. 63)

5.35.3.10 virtual SRMReturnCode Arc::SRMClient::mkDir (SRMClientRequest & *req*) [pure virtual]

Make required directories for the SURL in the request

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.11 virtual SRMReturnCode Arc::SRMClient::ping (std::string & *version*, bool *report_error* = true) [pure virtual]

Find out the version supported by the server this client is connected to. Since this method is used to determine which client version to instantiate, we may not want to report an error to the user, so setting *report_error* to false suppresses the error message.

Parameters:

version The version returned by the server
report_error Whether an error should be reported

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.12 SRMReturnCode Arc::SRMClient::process (PayloadSOAP * *request*, PayloadSOAP ** *response*) [protected]

Process SOAP request

5.35.3.13 virtual SRMReturnCode Arc::SRMClient::putTURLs (SRMClientRequest & *req*, std::list< std::string > & *urls*) [pure virtual]

If the user wishes to copy a file to somewhere, **putTURLs()** (p. 55) is called to retrieve the transport URL(s) to copy the file to. It may be used synchronously or asynchronously, depending on the synchronous property of the request object. In the former case it will block until the TURLs are ready, in the latter case it will return after making the request and **putTURLsStatus()** (p. 55) must be used to poll the request status if it was not completed.

Parameters:

req The request object

urls A list of TURLs filled by the method

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.14 virtual SRMReturnCode Arc::SRMClient::putTURLsStatus (SRMClientRequest & *req*, std::list< std::string > & *urls*) [pure virtual]

In the case where putTURLs was called asynchronously and the request was not completed, this method should be called to poll the status of the request. putTURLs must be called before this method and the request object must have ongoing request status.

Parameters:

req The request object. Status must be ongoing.

urls A list of TURLs filled by the method if the request completed successfully

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.15 virtual SRMReturnCode Arc::SRMClient::release (SRMClientRequest & *req*) [pure virtual]

Used in SRM v1 only. Called to release files after successful transfer.

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.16 virtual SRMReturnCode Arc::SRMClient::releaseGet (SRMClientRequest & *req*) [pure virtual]

Should be called after a successful copy from SRM storage.

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.17 virtual SRMReturnCode Arc::SRMClient::releasePut (SRMClientRequest & *req*)
[pure virtual]

Should be called after a successful copy to SRM storage.

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.18 virtual SRMReturnCode Arc::SRMClient::remove (SRMClientRequest & *req*) [pure
virtual]

Delete a file physically from storage and the SRM namespace.

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.19 virtual SRMReturnCode Arc::SRMClient::requestBringOnline (SRMClientRequest & *req*) [pure virtual]

Submit a request to bring online files. If the synchronous property of the request object is false, this operation is asynchronous and the status of the request can be checked by calling **requestBringOnlineStatus()** (p. 57) with the request token in *req* which is assigned by this method. If the request is synchronous, this operation blocks until the file(s) are online or the timeout specified in the **SRMClient** (p. 50) constructor has passed.

Parameters:

req The request object

Returns:

SRMReturnCode specifying outcome of operation

5.35.3.20 virtual SRMReturnCode Arc::SRMClient::requestBringOnlineStatus (SRMClientRequest & req) [pure virtual]

Query the status of a request to bring files online. The URLs map of the request object is updated if the status of any files in the request has changed. **requestBringOnline()** (p. 56) but be called before this method.

Parameters:

req The request object to query the status of

Returns:

SRMReturnCode specifying outcome of operation

5.35.4 Field Documentation

5.35.4.1 MCCCConfig Arc::SRMClient::cfg [protected]

SOAP configuraton object

5.35.4.2 ClientSOAP* Arc::SRMClient::client [protected]

SOAP client object

5.35.4.3 time_t Arc::SRMClient::connection_timeout [static, protected]

Timeout for connection to service

5.35.4.4 SRMImplementation Arc::SRMClient::implementation [protected]

The implementation of the server

5.35.4.5 Logger Arc::SRMClient::logger [static, protected]

Logger

5.35.4.6 NS Arc::SRMClient::ns [protected]

SOAP namespace

5.35.4.7 std::string Arc::SRMClient::service_endpoint [protected]

The URL of the service endpoint, eg <http://srm.ndgf.org:8443/srm/managerv2> All URLs passed to methods must correspond to this endpoint.

5.35.4.8 time_t Arc::SRMClient::user_timeout [protected]

Timeout for requests to the SRM service

5.35.4.9 `std::string Arc::SRMClient::version` `[protected]`

The version of the SRM protocol used

The documentation for this class was generated from the following file:

- SRMClient.h

5.36 Arc::SRMClientRequest Class Reference

```
#include <SRMClient.h>
```

Public Member Functions

- **SRMClientRequest** (const std::list< std::string > &urls) throw (SRMInvalidRequestException)
- **SRMClientRequest** (const std::string &url="", const std::string &id="") throw (SRMInvalidRequestException)
- **void request_id** (int id)
- **void request_token** (const std::string &token)
- **void file_ids** (const std::list< int > &ids)
- **void space_token** (const std::string &token)
- **std::list< std::string > surls** () const
- **void surl_statuses** (const std::string &surl, SRMFileLocality locality)
- **void surl_failures** (const std::string &surl, const std::string &reason)
- **void waiting_time** (int wait_time)
- **void finished_success** ()
- **void request_timeout** (unsigned int timeout)
- **void total_size** (unsigned long long size)
- **void long_list** (bool list)

5.36.1 Detailed Description

Class to represent a request which may be used for multiple operations, for example calling getTURLs() sets the request token in the request object (for a v2.2 client) and then same object is passed to releaseGet().

5.36.2 Constructor & Destructor Documentation

5.36.2.1 Arc::SRMClientRequest::SRMClientRequest (const std::list< std::string > &urls) throw (SRMInvalidRequestException) [inline]

Creates a request object with multiple SURLs. The URLs here are in the form srm://srm.ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/file3

5.36.2.2 Arc::SRMClientRequest::SRMClientRequest (const std::string &url = "", const std::string &id = "") throw (SRMInvalidRequestException) [inline]

Creates a request object with a single SURL. The URL here are in the form srm://srm.ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/file3

5.36.3 Member Function Documentation

5.36.3.1 void Arc::SRMClientRequest::file_ids (const std::list< int > &ids) [inline]

set and get file id list

5.36.3.2 void Arc::SRMClientRequest::finished_success () [inline]

set and get status of request

5.36.3.3 void Arc::SRMClientRequest::long_list (bool *list*) [inline]

set and get long list flag

5.36.3.4 void Arc::SRMClientRequest::request_id (int *id*) [inline]

set and get request id

5.36.3.5 void Arc::SRMClientRequest::request_timeout (unsigned int *timeout*) [inline]

set and get request timeout

5.36.3.6 void Arc::SRMClientRequest::request_token (const std::string & *token*) [inline]

set and get request token

5.36.3.7 void Arc::SRMClientRequest::space_token (const std::string & *token*) [inline]

set and get space token

5.36.3.8 void Arc::SRMClientRequest::surl_failures (const std::string & *surl*, const std::string & *reason*) [inline]

set and get surl failures

5.36.3.9 void Arc::SRMClientRequest::surl_statuses (const std::string & *surl*, SRMFileLocality *locality*) [inline]

set and get surl statuses

5.36.3.10 std::list<std::string> Arc::SRMClientRequest::surls () const [inline]

get URLs

5.36.3.11 void Arc::SRMClientRequest::total_size (unsigned long long *size*) [inline]

set and get total size

5.36.3.12 void Arc::SRMClientRequest::waiting_time (int *wait_time*) [inline]

set and get waiting time. A waiting time of zero means no estimate was given by the remote service.

The documentation for this class was generated from the following file:

- SRMClient.h

5.37 SRMFileInfo Class Reference

```
#include <SRMInfo.h>
```

5.37.1 Detailed Description

Info about a particular entry in the SRM info file

The documentation for this class was generated from the following file:

- SRMInfo.h

5.38 Arc::SRMFileMetaData Struct Reference

```
#include <SRMClient.h>
```

5.38.1 Detailed Description

File metadata

The documentation for this struct was generated from the following file:

- SRMClient.h

5.39 SRMInfo Class Reference

```
#include <SRMInfo.h>
```

5.39.1 Detailed Description

Represents SRM info stored in file. A combination of host and SRM version make a unique entry.

The documentation for this class was generated from the following file:

- SRMInfo.h

5.40 ArcSec::UsernameTokenSH Class Reference

Adds WS-Security Username Token into SOAP Header.

```
#include <UsernameTokenSH.h>
```

5.40.1 Detailed Description

Adds WS-Security Username Token into SOAP Header.

The documentation for this class was generated from the following file:

- UsernameTokenSH.h

5.41 ArcSec::X509TokenSH Class Reference

Adds WS-Security X509 Token into SOAP Header.

```
#include <X509TokenSH.h>
```

5.41.1 Detailed Description

Adds WS-Security X509 Token into SOAP Header.

The documentation for this class was generated from the following file:

- X509TokenSH.h

5.42 ArcSec::XACMLAlgFactory Class Reference

Algorithm factory class for XACML.

```
#include <XACMLAlgFactory.h>
```

Public Member Functions

- virtual CombiningAlg * createAlg (const std::string &type)

5.42.1 Detailed Description

Algorithm factory class for XACML.

5.42.2 Member Function Documentation

5.42.2.1 virtual CombiningAlg* ArcSec::XACMLAlgFactory::createAlg (const std::string & type)
[virtual]

return a Alg object according to the "CombiningAlg" attribute in the <Policy> node; The **XACMLAlgFactory** (p. 67) itself will release the Alg objects

The documentation for this class was generated from the following file:

- XACMLAlgFactory.h

5.43 ArcSec::XACMLAttributeFactory Class Reference

Attribute factory class for XACML specified attributes.

```
#include <XACMLAttributeFactory.h>
```

Public Member Functions

- virtual AttributeValue * **createValue** (const Arc::XMLNode &node, const std::string &type)

5.43.1 Detailed Description

Attribute factory class for XACML specified attributes.

5.43.2 Member Function Documentation

5.43.2.1 virtual AttributeValue* ArcSec::XACMLAttributeFactory::createValue (const Arc::XMLNode & node, const std::string & type) [virtual]

creat a AttributeValue according to the value in the XML node and the type; It should be the caller to release the AttributeValue Object

The documentation for this class was generated from the following file:

- XACMLAttributeFactory.h

5.44 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference

XACML specific AttributeProxy class.

```
#include <XACMLAttributeProxy.h>
```

Public Member Functions

- virtual AttributeValue * **getAttribute** (const Arc::XMLNode &node)

5.44.1 Detailed Description

template<class TheAttribute> class ArcSec::XACMLAttributeProxy< TheAttribute >

XACML specific AttributeProxy class.

5.44.2 Member Function Documentation

5.44.2.1 template<class TheAttribute> AttributeValue * ArcSec::XACMLAttributeProxy< TheAttribute >::getAttribute (const Arc::XMLNode & node) [virtual]

Implementation of getAttribute method.

The documentation for this class was generated from the following file:

- XACMLAttributeProxy.h

5.45 ArcSec::XACMLCondition Class Reference

XACMLCondition (p. 70) class to parse and operate XACML specific <Condition> node.

```
#include <XACMLCondition.h>
```

Public Member Functions

- **XACMLCondition** (Arc::XMLNode &node, EvaluatorContext *ctx)

5.45.1 Detailed Description

XACMLCondition (p. 70) class to parse and operate XACML specific <Condition> node.

5.45.2 Constructor & Destructor Documentation

5.45.2.1 ArcSec::XACMLCondition::XACMLCondition (Arc::XMLNode & *node*, EvaluatorContext * *ctx*)

Constructor -

The documentation for this class was generated from the following file:

- XACMLCondition.h

5.46 ArcSec::XACMLEvaluationCtx Class Reference

EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.

```
#include <XACMLEvaluationCtx.h>
```

Public Member Functions

- **XACMLEvaluationCtx (Request *request)**

5.46.1 Detailed Description

EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.

5.46.2 Constructor & Destructor Documentation

5.46.2.1 ArcSec::XACMLEvaluationCtx::XACMLEvaluationCtx (Request * *request*)

Construct a new EvaluationCtx based on the given request

The documentation for this class was generated from the following file:

- XACMLEvaluationCtx.h

5.47 ArcSec::XACMLEvaluator Class Reference

Execute the policy evaluation, based on the request and policy.

```
#include <XACMLEvaluator.h>
```

Public Member Functions

- virtual Response * **evaluate** (Request *request)

5.47.1 Detailed Description

Execute the policy evaluation, based on the request and policy.

5.47.2 Member Function Documentation

5.47.2.1 virtual Response* ArcSec::XACMLEvaluator::evaluate (Request * *request*) [virtual]

Evaluate the request based on the policy information inside PolicyStore

The documentation for this class was generated from the following file:

- XACMLEvaluator.h

5.48 ArcSec::XACMLFnFactory Class Reference

Function factory class for XACML specified attributes.

```
#include <XACMLFnFactory.h>
```

Public Member Functions

- virtual Function * **createFn** (const std::string &type)

5.48.1 Detailed Description

Function factory class for XACML specified attributes.

5.48.2 Member Function Documentation

5.48.2.1 virtual Function* ArcSec::XACMLFnFactory::createFn (const std::string & type) [virtual]

return a Function object according to the "Function" attribute in the XML node; The **XACMLFnFactory** (p. 73) itself will release the Function objects

The documentation for this class was generated from the following file:

- XACMLFnFactory.h

5.49 ArcSec::XACMLPDP Class Reference

XACMLPDP (p. 74) - PDP which can handle the XACML specific request and policy schema.

```
#include <XACMLPDP.h>
```

5.49.1 Detailed Description

XACMLPDP (p. 74) - PDP which can handle the XACML specific request and policy schema.

The documentation for this class was generated from the following file:

- XACMLPDP.h

5.50 ArcSec::XACMLPolicy Class Reference

XACMLPolicy (p. 75) class to parse and operate XACML specific <Policy> node.

```
#include <XACMLPolicy.h>
```

Public Member Functions

- **XACMLPolicy** (void)
- **XACMLPolicy** (const Arc::XMLNode node)
- **XACMLPolicy** (const Arc::XMLNode node, EvaluatorContext *ctx)
- virtual void make_policy ()

5.50.1 Detailed Description

XACMLPolicy (p. 75) class to parse and operate XACML specific <Policy> node.

5.50.2 Constructor & Destructor Documentation

5.50.2.1 ArcSec::XACMLPolicy::XACMLPolicy (void)

Constructor

5.50.2.2 ArcSec::XACMLPolicy::XACMLPolicy (const Arc::XMLNode node)

Constructor

5.50.2.3 ArcSec::XACMLPolicy::XACMLPolicy (const Arc::XMLNode node, EvaluatorContext *ctx)

Constructor -

5.50.3 Member Function Documentation

5.50.3.1 virtual void ArcSec::XACMLPolicy::make_policy () [virtual]

Parse XMLNode, and construct the low-level Rule object

The documentation for this class was generated from the following file:

- XACMLPolicy.h

5.51 ArcSec::XACMLRule Class Reference

XACMLRule (p. 76) class to parse XACML specific <Rule> node.

```
#include <XACMLRule.h>
```

5.51.1 Detailed Description

XACMLRule (p. 76) class to parse XACML specific <Rule> node.

The documentation for this class was generated from the following file:

- XACMLRule.h

5.52 ArcSec::XACMLTarget Class Reference

XACMLTarget (p. 77) class to parse and operate XACML specific <Target> node.

```
#include <XACMLTarget.h>
```

Public Member Functions

- **XACMLTarget** (Arc::XMLNode &node, EvaluatorContext *ctx)

5.52.1 Detailed Description

XACMLTarget (p. 77) class to parse and operate XACML specific <Target> node.

5.52.2 Constructor & Destructor Documentation

5.52.2.1 ArcSec::XACMLTarget::XACMLTarget (Arc::XMLNode & node, EvaluatorContext * ctx)

Constructor -

The documentation for this class was generated from the following file:

- XACMLTarget.h

Index

- ~LDAPQuery
 - Arc::LDAPQuery, 26
- ~PayloadTLSStream
 - Arc::PayloadTLSStream, 44
- ~SRMClient
 - Arc::SRMClient, 51
- abort
 - Arc::SRMClient, 51
- AndList
 - ArcSec, 10
- Arc::LDAPQuery, 26
 - ~LDAPQuery, 26
 - LDAPQuery, 26
 - Query, 26
 - Result, 26
- Arc::MCC_HTTP, 27
- Arc::MCC_HTTP_Client, 28
- Arc::MCC_HTTP_Service, 29
- Arc::MCC_SOAP, 30
- Arc::MCC_SOAP_Service, 31
- Arc::MCC_TCP, 32
- Arc::MCC_TCP_Client, 33
- Arc::MCC_TCP_Service, 34
 - MCC_TCP_Service, 34
- Arc::MCC_TLS, 35
- Arc::MCC_TLS_Client, 36
- Arc::MCC_TLS_Service, 37
- Arc::PayloadHTTP, 38
- Arc::PayloadHTTP
 - Attribute, 39
 - Attributes, 39
 - attributes_, 40
 - Body, 39
 - body_own_, 40
 - chunked_, 40
 - code_, 40
 - Flush, 40
 - get_body, 40
 - keep_alive_, 40
 - length_, 41
 - method_, 41
 - parse_header, 40
 - PayloadHTTP, 39
 - rbody_, 41
 - read, 40
 - readline, 40
 - reason_, 41
 - sbody_, 41
 - stream_, 41
 - stream_own_, 41
 - uri_, 41
 - version_major_, 41
 - version_minor_, 41
- Arc::PayloadTCPSocket, 43
 - Arc::PayloadTCPSocket
 - PayloadTCPSocket, 43
- Arc::PayloadTLSStream, 44
 - Arc::PayloadTLSStream
 - ~PayloadTLSStream, 44
 - GetCert, 44
 - GetPeerCert, 44
 - PayloadTLSStream, 44
 - ssl_, 45
 - STACK_OF, 44
- Arc::SRMClient, 50
 - ~SRMClient, 51
 - abort, 51
 - cfg, 57
 - client, 57
 - connection_timeout, 57
 - copy, 51
 - getInstance, 52
 - getRequestTokens, 52
 - getSpaceTokens, 52
 - getURLs, 53
 - getURLsStatus, 53
 - getVersion, 53
 - implementation, 57
 - info, 53
 - logger, 57
 - mkDir, 54
 - ns, 57
 - ping, 54
 - process, 54
 - putURLs, 54
 - putURLsStatus, 55
 - release, 55
 - releaseGet, 55
 - releasePut, 56

- remove, 56
- requestBringOnline, 56
- requestBringOnlineStatus, 56
- service_endpoint, 57
- SRMClient, 51
- user_timeout, 57
- version, 57
- Arc::SRMClientRequest, 59
- Arc::SRMClientRequest
 - file_ids, 59
 - finished_success, 59
 - long_list, 60
 - request_id, 60
 - request_timeout, 60
 - request_token, 60
 - space_token, 60
 - SRMClientRequest, 59
 - surl_failures, 60
 - surl_statuses, 60
 - surls, 60
 - total_size, 60
 - waiting_time, 60
- Arc::SRMFileMetaData, 63
- ArcEvaluationCtx
 - ArcSec::ArcEvaluationCtx, 16
- ArcPolicy
 - ArcSec::ArcPolicy, 20
- ArcSec, 7
- ArcSec
 - AndList, 10
 - Match, 10
 - OrList, 10
- ArcSec::AllowPDP, 11
- ArcSec::ArcAlgFactory, 12
- ArcSec::ArcAlgFactory
 - createAlg, 12
- ArcSec::ArcAttributeFactory, 13
- ArcSec::ArcAttributeFactory
 - createValue, 13
- ArcSec::ArcAttributeProxy, 14
- ArcSec::ArcAttributeProxy
 - getAttribute, 14
- ArcSec::ArcAuthZ, 15
- ArcSec::ArcAuthZ
 - Handle, 15
 - MakePDPs, 15
- ArcSec::ArcEvaluationCtx, 16
- ArcSec::ArcEvaluationCtx
 - ArcEvaluationCtx, 16
 - split, 16
- ArcSec::ArcEvaluator, 17
- ArcSec::ArcEvaluator
 - evaluate, 17
- ArcSec::ArcFnFactory, 18
- ArcSec::ArcFnFactory
 - createFn, 18
- ArcSec::ArcPDP, 19
- ArcSec::ArcPolicy, 20
- ArcSec::ArcPolicy
 - ArcPolicy, 20
 - make_policy, 20
- ArcSec::ArcRequestItem, 21
- ArcSec::ArcRequestTuple, 22
- ArcSec::ArcRule, 23
- ArcSec::DelegationPDP, 24
- ArcSec::DenyPDP, 25
- ArcSec::PDPServiceInvoker, 46
- ArcSec::SAML2SSO_AssertionConsumerSH, 47
- ArcSec::SAMLTokenSH, 48
- ArcSec::SimpleListPDP, 49
- ArcSec::UsernameTokenSH, 65
- ArcSec::X509TokenSH, 66
- ArcSec::XACMLAlgFactory, 67
- ArcSec::XACMLAlgFactory
 - createAlg, 67
- ArcSec::XACMLAttributeFactory, 68
- ArcSec::XACMLAttributeFactory
 - createValue, 68
- ArcSec::XACMLAttributeProxy, 69
- ArcSec::XACMLAttributeProxy
 - getAttribute, 69
- ArcSec::XACMLCondition, 70
- ArcSec::XACMLCondition
 - XACMLCondition, 70
- ArcSec::XACMLEvaluationCtx, 71
- ArcSec::XACMLEvaluationCtx
 - XACMLEvaluationCtx, 71
- ArcSec::XACMLEvaluator, 72
- ArcSec::XACMLEvaluator
 - evaluate, 72
- ArcSec::XACMLFnFactory, 73
- ArcSec::XACMLFnFactory
 - createFn, 73
- ArcSec::XACMLPDP, 74
- ArcSec::XACMLPolicy, 75
- ArcSec::XACMLPolicy
 - make_policy, 75
 - XACMLPolicy, 75
- ArcSec::XACMLRule, 76
- ArcSec::XACMLTarget, 77
- ArcSec::XACMLTarget
 - XACMLTarget, 77
- Attribute
 - Arc::PayloadHTTP, 39
- Attributes
 - Arc::PayloadHTTP, 39
- attributes_
 - Arc::PayloadHTTP, 40

- Body
 - Arc::PayloadHTTP, 39
- body_own_
 - Arc::PayloadHTTP, 40
- cfg
 - Arc::SRMClient, 57
- chunked_
 - Arc::PayloadHTTP, 40
- client
 - Arc::SRMClient, 57
- code_
 - Arc::PayloadHTTP, 40
- connection_timeout
 - Arc::SRMClient, 57
- copy
 - Arc::SRMClient, 51
- createAlg
 - ArcSec::ArcAlgFactory, 12
 - ArcSec::XACMLAlgFactory, 67
- createFn
 - ArcSec::ArcFnFactory, 18
 - ArcSec::XACMLFnFactory, 73
- createValue
 - ArcSec::ArcAttributeFactory, 13
 - ArcSec::XACMLAttributeFactory, 68
- evaluate
 - ArcSec::ArcEvaluator, 17
 - ArcSec::XACMLEvaluator, 72
- file_ids
 - Arc::SRMClientRequest, 59
- finished_success
 - Arc::SRMClientRequest, 59
- Flush
 - Arc::PayloadHTTP, 40
- get_body
 - Arc::PayloadHTTP, 40
- getAttribute
 - ArcSec::ArcAttributeProxy, 14
 - ArcSec::XACMLAttributeProxy, 69
- GetCert
 - Arc::PayloadTLSStream, 44
- getInstance
 - Arc::SRMClient, 52
- GetPeerCert
 - Arc::PayloadTLSStream, 44
- getRequestTokens
 - Arc::SRMClient, 52
- getSpaceTokens
 - Arc::SRMClient, 52
- getURLs
 - Arc::SRMClient, 53
- getURLsStatus
 - Arc::SRMClient, 53
- getVersion
 - Arc::SRMClient, 53
- Handle
 - ArcSec::ArcAuthZ, 15
- implementation
 - Arc::SRMClient, 57
- info
 - Arc::SRMClient, 53
- keep_alive_
 - Arc::PayloadHTTP, 40
- LDAPQuery
 - Arc::LDAPQuery, 26
- length_
 - Arc::PayloadHTTP, 41
- logger
 - Arc::SRMClient, 57
- long_list
 - Arc::SRMClientRequest, 60
- make_policy
 - ArcSec::ArcPolicy, 20
 - ArcSec::XACMLPolicy, 75
- MakePDPs
 - ArcSec::ArcAuthZ, 15
- Match
 - ArcSec, 10
- MCC_TCP_Service
 - Arc::MCC_TCP_Service, 34
- method_
 - Arc::PayloadHTTP, 41
- mkDir
 - Arc::SRMClient, 54
- ns
 - Arc::SRMClient, 57
- OrList
 - ArcSec, 10
- parse_header
 - Arc::PayloadHTTP, 40
- PayloadHTTP
 - Arc::PayloadHTTP, 39
- PayloadTCPSocket
 - Arc::PayloadTCPSocket, 43
- PayloadTLSStream
 - Arc::PayloadTLSStream, 44
- ping

- Arc::SRMClient, 54
- process
 - Arc::SRMClient, 54
- putURLs
 - Arc::SRMClient, 54
- putURLsStatus
 - Arc::SRMClient, 55
- Query
 - Arc::LDAPQuery, 26
- rbody_
 - Arc::PayloadHTTP, 41
- read
 - Arc::PayloadHTTP, 40
- readline
 - Arc::PayloadHTTP, 40
- reason_
 - Arc::PayloadHTTP, 41
- release
 - Arc::SRMClient, 55
- releaseGet
 - Arc::SRMClient, 55
- releasePut
 - Arc::SRMClient, 56
- remove
 - Arc::SRMClient, 56
- request_id
 - Arc::SRMClientRequest, 60
- request_timeout
 - Arc::SRMClientRequest, 60
- request_token
 - Arc::SRMClientRequest, 60
- requestBringOnline
 - Arc::SRMClient, 56
- requestBringOnlineStatus
 - Arc::SRMClient, 56
- Result
 - Arc::LDAPQuery, 26
- sbody_
 - Arc::PayloadHTTP, 41
- service_endpoint
 - Arc::SRMClient, 57
- space_token
 - Arc::SRMClientRequest, 60
- split
 - ArcSec::ArcEvaluationCtx, 16
- SRMClient
 - Arc::SRMClient, 51
- SRMClientRequest
 - Arc::SRMClientRequest, 59
- SRMFileInfo, 62
- SRMInfo, 64
- ssl_
 - Arc::PayloadTLSStream, 45
- STACK_OF
 - Arc::PayloadTLSStream, 44
- stream_
 - Arc::PayloadHTTP, 41
- stream_own_
 - Arc::PayloadHTTP, 41
- surl_failures
 - Arc::SRMClientRequest, 60
- surl_statuses
 - Arc::SRMClientRequest, 60
- surls
 - Arc::SRMClientRequest, 60
- total_size
 - Arc::SRMClientRequest, 60
- uri_
 - Arc::PayloadHTTP, 41
- user_timeout
 - Arc::SRMClient, 57
- version
 - Arc::SRMClient, 57
- version_major_
 - Arc::PayloadHTTP, 41
- version_minor_
 - Arc::PayloadHTTP, 41
- waiting_time
 - Arc::SRMClientRequest, 60
- XACMLCondition
 - ArcSec::XACMLCondition, 70
- XACMLEvaluationCtx
 - ArcSec::XACMLEvaluationCtx, 71
- XACMLPolicy
 - ArcSec::XACMLPolicy, 75
- XACMLTarget
 - ArcSec::XACMLTarget, 77