

Hosting Environment (Daemon) Chain Components

Generated by Doxygen 1.7.1

Thu Dec 9 2010 14:24:34

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Data Structure Index	3
2.1	Class Hierarchy	3
3	Data Structure Index	7
3.1	Data Structures	7
4	Namespace Documentation	11
4.1	ArcSec Namespace Reference	11
4.1.1	Detailed Description	13
4.1.2	Typedef Documentation	14
4.1.2.1	AndList	14
4.1.2.2	Match	14
5	Data Structure Documentation	15
5.1	ArcSec::AllowPDP Class Reference	15
5.1.1	Detailed Description	15
5.2	ArcSec::ArcAlgFactory Class Reference	15
5.2.1	Detailed Description	15
5.2.2	Member Function Documentation	16
5.2.2.1	createAlg	16
5.3	ArcSec::ArcAttributeFactory Class Reference	16
5.3.1	Detailed Description	16
5.3.2	Member Function Documentation	16
5.3.2.1	createValue	16
5.4	ArcSec::ArcAttributeProxy< TheAttribute > Class Template Reference	16
5.4.1	Detailed Description	17
5.5	ArcSec::ArcAuthZ Class Reference	17

5.5.1	Detailed Description	17
5.5.2	Member Function Documentation	17
5.5.2.1	Handle	17
5.5.2.2	MakePDPs	17
5.6	ArcSec::ArcEvaluationCtx Class Reference	18
5.6.1	Detailed Description	18
5.6.2	Constructor & Destructor Documentation	18
5.6.2.1	ArcEvaluationCtx	18
5.6.3	Member Function Documentation	18
5.6.3.1	split	18
5.7	ArcSec::ArcEvaluator Class Reference	18
5.7.1	Detailed Description	18
5.7.2	Member Function Documentation	19
5.7.2.1	evaluate	19
5.8	ArcSec::ArcFnFactory Class Reference	19
5.8.1	Detailed Description	19
5.8.2	Member Function Documentation	19
5.8.2.1	createFn	19
5.9	ArcSec::ArcPDP Class Reference	19
5.9.1	Detailed Description	19
5.10	ArcSec::ArcPolicy Class Reference	20
5.10.1	Detailed Description	20
5.10.2	Constructor & Destructor Documentation	20
5.10.2.1	ArcPolicy	20
5.10.2.2	ArcPolicy	20
5.10.2.3	ArcPolicy	20
5.10.3	Member Function Documentation	20
5.10.3.1	make_policy	20
5.11	ArcSec::ArcRequest Class Reference	20
5.12	ArcSec::ArcRequestItem Class Reference	21
5.12.1	Detailed Description	21
5.13	ArcSec::ArcRequestTuple Class Reference	21
5.13.1	Detailed Description	21
5.14	ArcSec::ArcRule Class Reference	21
5.14.1	Detailed Description	21
5.15	ArcSec::AttributeDesignator Class Reference	21

5.16	ArcSec::AttributeSelector Class Reference	22
5.17	Arc::ConfigTLSMCC Class Reference	22
5.18	Arc::DataPointARC Class Reference	22
5.19	Arc::DataPointFile Class Reference	22
5.20	Arc::DataPointGridFTP Class Reference	22
5.21	Arc::DataPointHTTP Class Reference	22
5.22	Arc::DataPointLDAP Class Reference	22
5.23	Arc::DataPointLFC Class Reference	23
5.24	Arc::DataPointRLS Class Reference	23
5.25	Arc::DataPointSRM Class Reference	23
5.26	ArcSec::DelegationCollector Class Reference	23
5.27	ArcSec::DelegationMultiSecAttr Class Reference	23
5.28	ArcSec::DelegationPDP Class Reference	23
5.28.1	Detailed Description	23
5.29	ArcSec::DelegationSecAttr Class Reference	24
5.30	ArcSec::DelegationSH Class Reference	24
5.31	ArcSec::DenyPDP Class Reference	24
5.31.1	Detailed Description	24
5.32	ArcSec::GACLEvaluator Class Reference	24
5.32.1	Member Function Documentation	24
5.32.1.1	evaluate	24
5.33	ArcSec::GACLPDP Class Reference	24
5.34	ArcSec::GACLPolicy Class Reference	25
5.35	ArcSec::GACLRequest Class Reference	25
5.36	Arc::LDAPQuery Class Reference	25
5.36.1	Detailed Description	25
5.36.2	Constructor & Destructor Documentation	25
5.36.2.1	LDAPQuery	25
5.36.2.2	~LDAPQuery	25
5.36.3	Member Function Documentation	26
5.36.3.1	Query	26
5.36.3.2	Result	26
5.37	Arc::Lister Class Reference	26
5.38	Arc::MCC_GSI_Client Class Reference	26
5.39	Arc::MCC_GSI_Service Class Reference	26
5.40	Arc::MCC_HTTP Class Reference	26

5.40.1 Detailed Description	27
5.41 Arc::MCC_HTTP_Client Class Reference	27
5.41.1 Detailed Description	27
5.42 Arc::MCC_HTTP_Service Class Reference	27
5.42.1 Detailed Description	28
5.43 Arc::MCC_MsgValidator Class Reference	28
5.44 Arc::MCC_MsgValidator_Service Class Reference	28
5.45 Arc::MCC_SOAP Class Reference	28
5.45.1 Detailed Description	29
5.46 Arc::MCC_SOAP_Client Class Reference	29
5.47 Arc::MCC_SOAP_Service Class Reference	29
5.47.1 Detailed Description	30
5.48 Arc::MCC_TCP Class Reference	30
5.48.1 Detailed Description	30
5.49 Arc::MCC_TCP_Client Class Reference	30
5.49.1 Detailed Description	30
5.50 Arc::MCC_TCP_Service Class Reference	31
5.50.1 Detailed Description	31
5.50.2 Constructor & Destructor Documentation	31
5.50.2.1 MCC_TCP_Service	31
5.51 Arc::MCC_TLS Class Reference	32
5.51.1 Detailed Description	32
5.52 Arc::MCC_TLS_Client Class Reference	32
5.52.1 Detailed Description	32
5.53 Arc::MCC_TLS_Service Class Reference	32
5.53.1 Detailed Description	33
5.54 Arc::PayloadGSISStream Class Reference	33
5.55 Arc::PayloadHTTP Class Reference	33
5.55.1 Detailed Description	34
5.55.2 Constructor & Destructor Documentation	34
5.55.2.1 PayloadHTTP	34
5.55.2.2 PayloadHTTP	34
5.55.2.3 PayloadHTTP	34
5.55.2.4 PayloadHTTP	35
5.55.2.5 PayloadHTTP	35
5.55.3 Member Function Documentation	35

5.55.3.1	Attribute	35
5.55.3.2	Attribute	35
5.55.3.3	Attributes	35
5.55.3.4	Body	35
5.55.3.5	Flush	35
5.55.3.6	get_body	35
5.55.3.7	parse_header	35
5.55.3.8	read	36
5.55.3.9	readline	36
5.55.4	Field Documentation	36
5.55.4.1	attributes_	36
5.55.4.2	body_own_	36
5.55.4.3	chunked_	36
5.55.4.4	code_	36
5.55.4.5	keep_alive_	36
5.55.4.6	length_	36
5.55.4.7	method_	36
5.55.4.8	rbody_	36
5.55.4.9	reason_	36
5.55.4.10	sbody_	37
5.55.4.11	stream_	37
5.55.4.12	stream_own_	37
5.55.4.13	uri_	37
5.55.4.14	version_major_	37
5.55.4.15	version_minor_	37
5.56	Arc::PayloadTCPSocket Class Reference	37
5.56.1	Detailed Description	37
5.56.2	Constructor & Destructor Documentation	38
5.56.2.1	PayloadTCPSocket	38
5.56.2.2	PayloadTCPSocket	38
5.56.2.3	PayloadTCPSocket	38
5.56.2.4	PayloadTCPSocket	38
5.56.2.5	PayloadTCPSocket	38
5.57	Arc::PayloadTLMCC Class Reference	38
5.57.1	Constructor & Destructor Documentation	39
5.57.1.1	PayloadTLMCC	39

5.57.1.2	PayloadTLMCC	39
5.57.1.3	PayloadTLMCC	39
5.58	Arc::PayloadTLSSStream Class Reference	39
5.58.1	Detailed Description	40
5.58.2	Constructor & Destructor Documentation	40
5.58.2.1	PayloadTLSSStream	40
5.58.2.2	~PayloadTLSSStream	40
5.58.3	Member Function Documentation	40
5.58.3.1	GetCert	40
5.58.3.2	GetPeerCert	40
5.58.3.3	STACK_OF	40
5.58.4	Field Documentation	40
5.58.4.1	ssl_	40
5.59	ArcSec::PDPSERVICEInvoker Class Reference	40
5.59.1	Detailed Description	41
5.60	ArcSec::SAML2SSO_AssertionConsumerSH Class Reference	41
5.60.1	Detailed Description	41
5.61	ArcSec::SAMLTokenSH Class Reference	41
5.61.1	Detailed Description	41
5.62	ArcSec::SimpleListPDP Class Reference	41
5.62.1	Detailed Description	41
5.63	Arc::SRM1Client Class Reference	42
5.63.1	Member Function Documentation	42
5.63.1.1	abort	42
5.63.1.2	copy	43
5.63.1.3	getRequestTokens	43
5.63.1.4	getSpaceTokens	43
5.63.1.5	getURLs	44
5.63.1.6	info	44
5.63.1.7	mkDir	44
5.63.1.8	ping	45
5.63.1.9	putURLs	45
5.63.1.10	release	45
5.63.1.11	releaseGet	46
5.63.1.12	releasePut	46
5.63.1.13	remove	46

5.63.1.14 requestBringOnline	46
5.63.1.15 requestBringOnlineStatus	47
5.64 Arc::SRM22Client Class Reference	47
5.64.1 Constructor & Destructor Documentation	48
5.64.1.1 SRM22Client	48
5.64.1.2 ~SRM22Client	48
5.64.2 Member Function Documentation	48
5.64.2.1 abort	48
5.64.2.2 copy	48
5.64.2.3 getRequestTokens	48
5.64.2.4 getSpaceTokens	48
5.64.2.5 getTURLs	49
5.64.2.6 info	49
5.64.2.7 mkdir	49
5.64.2.8 ping	49
5.64.2.9 putTURLs	49
5.64.2.10 release	49
5.64.2.11 releaseGet	49
5.64.2.12 releasePut	50
5.64.2.13 remove	50
5.64.2.14 requestBringOnline	50
5.64.2.15 requestBringOnlineStatus	50
5.65 Arc::SRMClient Class Reference	50
5.65.1 Detailed Description	51
5.65.2 Constructor & Destructor Documentation	52
5.65.2.1 SRMClient	52
5.65.2.2 ~SRMClient	52
5.65.3 Member Function Documentation	52
5.65.3.1 abort	52
5.65.3.2 copy	52
5.65.3.3 getInstance	52
5.65.3.4 getRequestTokens	53
5.65.3.5 getSpaceTokens	53
5.65.3.6 getTURLs	53
5.65.3.7 getVersion	54
5.65.3.8 info	54

5.65.3.9	mkDir	54
5.65.3.10	ping	54
5.65.3.11	process	55
5.65.3.12	putURLs	55
5.65.3.13	release	55
5.65.3.14	releaseGet	55
5.65.3.15	releasePut	56
5.65.3.16	remove	56
5.65.3.17	requestBringOnline	56
5.65.3.18	requestBringOnlineStatus	57
5.65.3.19	Timeout	57
5.65.4	Field Documentation	57
5.65.4.1	cfg	57
5.65.4.2	client	57
5.65.4.3	implementation	57
5.65.4.4	logger	57
5.65.4.5	ns	57
5.65.4.6	request_timeout	57
5.65.4.7	service_endpoint	58
5.65.4.8	user_timeout	58
5.65.4.9	version	58
5.66	Arc::SRMClientRequest Class Reference	58
5.66.1	Detailed Description	58
5.66.2	Constructor & Destructor Documentation	59
5.66.2.1	SRMClientRequest	59
5.66.2.2	SRMClientRequest	59
5.66.3	Member Function Documentation	59
5.66.3.1	file_ids	59
5.66.3.2	finished_success	59
5.66.3.3	long_list	59
5.66.3.4	request_id	59
5.66.3.5	request_token	59
5.66.3.6	space_token	59
5.66.3.7	surl_failures	59
5.66.3.8	surl_statuses	60
5.66.3.9	surls	60

5.66.3.10 waiting_time	60
5.67 SRMFileInfo Class Reference	60
5.67.1 Detailed Description	60
5.68 Arc::SRMFileMetaData Struct Reference	60
5.68.1 Detailed Description	60
5.69 SRMInfo Class Reference	60
5.69.1 Detailed Description	61
5.70 Arc::SRMInvalidRequestException Class Reference	61
5.71 SRMURL Class Reference	61
5.71.1 Constructor & Destructor Documentation	61
5.71.1.1 SRMURL	61
5.71.2 Member Function Documentation	61
5.71.2.1 BaseURL	61
5.71.2.2 ContactURL	61
5.71.2.3 Endpoint	61
5.71.2.4 FileName	62
5.71.2.5 FullURL	62
5.71.2.6 PortDefined	62
5.71.2.7 SetSRMVersion	62
5.71.2.8 ShortURL	62
5.72 ArcSec::UsernameTokenSH Class Reference	62
5.72.1 Detailed Description	62
5.73 ArcSec::X509TokenSH Class Reference	62
5.73.1 Detailed Description	63
5.74 ArcSec::XACMLAlgFactory Class Reference	63
5.74.1 Detailed Description	63
5.74.2 Member Function Documentation	63
5.74.2.1 createAlg	63
5.75 ArcSec::XACMLApply Class Reference	63
5.76 ArcSec::XACMLAttributeFactory Class Reference	63
5.76.1 Detailed Description	64
5.76.2 Member Function Documentation	64
5.76.2.1 createValue	64
5.77 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference	64
5.77.1 Detailed Description	64
5.78 ArcSec::XACMLCondition Class Reference	64

5.78.1 Detailed Description	65
5.78.2 Constructor & Destructor Documentation	65
5.78.2.1 XACMLCondition	65
5.79 ArcSec::XACMLEvaluationCtx Class Reference	65
5.79.1 Detailed Description	65
5.79.2 Constructor & Destructor Documentation	65
5.79.2.1 XACMLEvaluationCtx	65
5.80 ArcSec::XACMLEvaluator Class Reference	65
5.80.1 Detailed Description	66
5.80.2 Member Function Documentation	66
5.80.2.1 evaluate	66
5.81 ArcSec::XACMLFnFactory Class Reference	66
5.81.1 Detailed Description	66
5.81.2 Member Function Documentation	66
5.81.2.1 createFn	66
5.82 ArcSec::XACMLPDP Class Reference	67
5.82.1 Detailed Description	67
5.83 ArcSec::XACMLPolicy Class Reference	67
5.83.1 Detailed Description	67
5.83.2 Constructor & Destructor Documentation	67
5.83.2.1 XACMLPolicy	67
5.83.2.2 XACMLPolicy	67
5.83.2.3 XACMLPolicy	67
5.83.3 Member Function Documentation	68
5.83.3.1 make_policy	68
5.84 ArcSec::XACMLRequest Class Reference	68
5.84.1 Member Function Documentation	68
5.84.1.1 getEvalName	68
5.84.1.2 getName	68
5.85 ArcSec::XACMLRule Class Reference	68
5.85.1 Detailed Description	68
5.86 ArcSec::XACMLTarget Class Reference	69
5.86.1 Detailed Description	69
5.86.2 Constructor & Destructor Documentation	69
5.86.2.1 XACMLTarget	69
5.87 ArcSec::XACMLTargetMatch Class Reference	69

5.88 ArcSec::XACMLTargetMatchGroup Class Reference	69
5.89 ArcSec::XACMLTargetSection Class Reference	69

Chapter 1

Namespace Index

1.1 Namespace List

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

ArcSec (ArcRequest (p. 20), Parsing the specified Arc request format)	11
-------------------------------------------------------------------------------------------------	----

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

ArcSec::AllowPDP	15
ArcSec::ArcAlgFactory	15
ArcSec::ArcAttributeFactory	16
ArcSec::ArcAttributeProxy< TheAttribute >	16
ArcSec::ArcAuthZ	17
ArcSec::ArcEvaluationCtx	18
ArcSec::ArcEvaluator	18
ArcSec::ArcFnFactory	19
ArcSec::ArcPDP	19
ArcSec::ArcPolicy	20
ArcSec::ArcRequest	20
ArcSec::ArcRequestItem	21
ArcSec::ArcRequestTuple	21
ArcSec::ArcRule	21
ArcSec::AttributeDesignator	21
ArcSec::AttributeSelector	22
Arc::ConfigTlsmcc	22
Arc::DataPointARC	22
Arc::DataPointFile	22
Arc::DataPointGridFTP	22
Arc::DataPointHTTP	22
Arc::DataPointLDAP	22
Arc::DataPointLFC	23
Arc::DataPointRLS	23
Arc::DataPointSRM	23
ArcSec::DelegationCollector	23
ArcSec::DelegationMultiSecAttr	23
ArcSec::DelegationPDP	23
ArcSec::DelegationSecAttr	24
ArcSec::DelegationSH	24
ArcSec::DenyPDP	24
ArcSec::GACLEvaluator	24
ArcSec::GACLPDP	24

ArcSec::GACLPolicy	25
ArcSec::GACLRequest	25
Arc::LDAPQuery	25
Arc::Lister	26
Arc::MCC_GSI_Client	26
Arc::MCC_GSI_Service	26
Arc::MCC_HTTP	26
Arc::MCC_HTTP_Client	27
Arc::MCC_HTTP_Service	27
Arc::MCC_MsgValidator	28
Arc::MCC_MsgValidator_Service	28
Arc::MCC_SOAP	28
Arc::MCC_SOAP_Client	29
Arc::MCC_SOAP_Service	29
Arc::MCC_TCP	30
Arc::MCC_TCP_Client	30
Arc::MCC_TCP_Service	31
Arc::MCC_TLS	32
Arc::MCC_TLS_Client	32
Arc::MCC_TLS_Service	32
Arc::PayloadGSISStream	33
Arc::PayloadHTTP	33
Arc::PayloadTCPSocket	37
Arc::PayloadTLSSStream	39
Arc::PayloadTLSMCC	38
ArcSec::PDPServiceInvoker	40
ArcSec::SAML2SSO_AssertionConsumerSH	41
ArcSec::SAMLTokenSH	41
ArcSec::SimpleListPDP	41
Arc::SRMClient	50
Arc::SRM1Client	42
Arc::SRM22Client	47
Arc::SRMClientRequest	58
SRMFileInfo	60
Arc::SRMFileMetaData	60
SRMInfo	60
Arc::SRMInvalidRequestException	61
SRMURL	61
ArcSec::UsernameTokenSH	62
ArcSec::X509TokenSH	62
ArcSec::XACMLAlgFactory	63
ArcSec::XACMLApply	63
ArcSec::XACMLAttributeFactory	63
ArcSec::XACMLAttributeProxy< TheAttribute >	64
ArcSec::XACMLCondition	64
ArcSec::XACMLEvaluationCtx	65
ArcSec::XACMLEvaluator	65
ArcSec::XACMLFnFactory	66
ArcSec::XACMLPDP	67
ArcSec::XACMLPolicy	67
ArcSec::XACMLRequest	68
ArcSec::XACMLRule	68

ArcSec::XACMLTarget	69
ArcSec::XACMLTargetMatch	69
ArcSec::XACMLTargetMatchGroup	69
ArcSec::XACMLTargetSection	69

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

ArcSec::AllowPDP (This PDP always return true (allow))	15
ArcSec::ArcAlgFactory (Algorithm factory class for Arc)	15
ArcSec::ArcAttributeFactory (Attribute factory class for Arc specified attributes)	16
ArcSec::ArcAttributeProxy < TheAttribute > (Arc specific AttributeProxy class)	16
ArcSec::ArcAuthZ (Tests message against list of PDPs)	17
ArcSec::ArcEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	18
ArcSec::ArcEvaluator (Execute the policy evaluation, based on the request and policy)	18
ArcSec::ArcFnFactory (Function factory class for Arc specified attributes)	19
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::ArcRequest	20
ArcSec::ArcRequestItem (Container, <Subjects, Actions, Objects, Contexts> tuple)	21
ArcSec::ArcRequestTuple (RequestTuple, container which includes the)	21
ArcSec::ArcRule (ArcRule (p. 21) class to parse Arc specific <Rule> node)	21
ArcSec::AttributeDesignator	21
ArcSec::AttributeSelector	22
Arc::ConfigTlsmcc	22
Arc::DataPointARC	22
Arc::DataPointFile	22
Arc::DataPointGridFTP	22
Arc::DataPointHTTP	22
Arc::DataPointLDAP	22
Arc::DataPointLFC	23
Arc::DataPointRLS	23
Arc::DataPointSRM	23
ArcSec::DelegationCollector	23
ArcSec::DelegationMultiSecAttr	23
ArcSec::DelegationPDP	23
ArcSec::DelegationSecAttr	24
ArcSec::DelegationSH	24
ArcSec::DenyPDP (This PDP always returns false (deny))	24

ArcSec::GACLEvaluator	24
ArcSec::GACLPDP	24
ArcSec::GACLPolicy	25
ArcSec::GACLRequest	25
Arc::LDAPQuery	25
Arc::Lister	26
Arc::MCC_GSI_Client	26
Arc::MCC_GSI_Service	26
Arc::MCC_HTTP (A base class for HTTP client and service MCCs)	26
Arc::MCC_HTTP_Client	27
Arc::MCC_HTTP_Service	27
Arc::MCC_MsgValidator	28
Arc::MCC_MsgValidator_Service	28
Arc::MCC_SOAP (A base class for SOAP client and service MCCs)	28
Arc::MCC_SOAP_Client	29
Arc::MCC_SOAP_Service	29
Arc::MCC_TCP (A base class for TCP client and service MCCs)	30
Arc::MCC_TCP_Client	30
Arc::MCC_TCP_Service	31
Arc::MCC_TLS (A base class for TLS client and service MCCs)	32
Arc::MCC_TLS_Client	32
Arc::MCC_TLS_Service	32
Arc::PayloadGSISStream	33
Arc::PayloadHTTP	33
Arc::PayloadTCPSocket	37
Arc::PayloadTLMCC	38
Arc::PayloadTLSStream	39
ArcSec::PDPSERVICEInvoker (PDPSERVICEInvoker (p. 40) - client which will invoke pdpservice)	40
ArcSec::SAML2SSO_AssertionConsumerSH (Implement the funcionality of the Service Provider in SAML2 SSO profile)	41
ArcSec::SAMLTokenSH (Adds WS-Security SAML Token into SOAP Header)	41
ArcSec::SimpleListPDP (Tests X509 subject against list of subjects in file)	41
Arc::SRM1Client	42
Arc::SRM22Client	47
Arc::SRMClient	50
Arc::SRMClientRequest	58
SRMFileInfo	60
Arc::SRMFileMetaData	60
SRMInfo	60
Arc::SRMInvalidRequestException	61
SRMURL	61
ArcSec::UsernameTokenSH (Adds WS-Security Username Token into SOAP Header)	62
ArcSec::X509TokenSH (Adds WS-Security X509 Token into SOAP Header)	62
ArcSec::XACMLAlgFactory (Algorithm factory class for XACML)	63
ArcSec::XACMLApply	63
ArcSec::XACMLAttributeFactory (Attribute factory class for XACML specified attributes) .	63
ArcSec::XACMLAttributeProxy < TheAttribute > (XACML specific AttributeProxy class) .	64
ArcSec::XACMLCondition (XACMLCondition (p. 64) class to parse and operate XACML specific <Condition> node)	64
ArcSec::XACMLEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	65
ArcSec::XACMLEvaluator (Execute the policy evaluation, based on the request and policy) .	65
ArcSec::XACMLFnFactory (Function factory class for XACML specified attributes)	66

ArcSec::XACMLPDP (XACMLPDP (p. 67) - PDP which can handle the XACML specific request and policy schema)	67
ArcSec::XACMLPolicy (XACMLPolicy (p. 67) class to parse and operate XACML specific <Policy> node)	67
ArcSec::XACMLRequest	68
ArcSec::XACMLRule (XACMLRule (p. 68) class to parse XACML specific <Rule> node) .	68
ArcSec::XACMLTarget (XACMLTarget (p. 69) class to parse and operate XACML specific <Target> node)	69
ArcSec::XACMLTargetMatch	69
ArcSec::XACMLTargetMatchGroup	69
ArcSec::XACMLTargetSection	69

Chapter 4

Namespace Documentation

4.1 ArcSec Namespace Reference

ArcRequest (p. 20), 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. 21) 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. 40) - 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. 64) 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. 67) - PDP which can handle the XACML specific request and policy schema.
- class **XACMLPolicy**
XACMLPolicy (p. 67) class to parse and operate XACML specific <Policy> node.
- class **XACMLRequest**
- class **XACMLRule**
XACMLRule (p. 68) class to parse XACML specific <Rule> node.
- class **XACMLTargetMatch**
- class **XACMLTargetMatchGroup**
- class **XACMLTargetSection**
- class **XACMLTarget**
XACMLTarget (p. 69) 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 (p. 20), Parsing the specified Arc request format. **XACMLRequest** (p. 68), Parsing the xacml request format.

4.1.2 Typedef Documentation

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

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

"And" relationship means the request should satisfy all of the items <Subject>
 <SubFraction type="X500DN">/O=Grid/OU=KnowARC/CN=XYZ</SubFraction>
 <SubFraction type="ShibName">urn:mace:shibboleth:examples</SubFraction> </Subject>
 "Or" relationship means 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>

4.1.2.2 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/usergroupA</Subject> Subjects> example inside <Rule>: <Subjects> <Subject type="X500Name">/O=NorduGrid/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>

Chapter 5

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. 15) 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.

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>
```

Data Structures

- class PDPDesc

Public Member Functions

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

Protected Member Functions

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

5.5.1 Detailed Description

Tests message against list of PDPs. This class implements SecHandler interface. It's **Handle()** (p. 17) method runs provided Message instance against all PDPs specified in configuration. If any of PDPs returns positive result **Handle()** (p. 17) 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 RequestTuple 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. 19) 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::ArcRequest Class Reference

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

- ArcRequest.h

5.12 ArcSec::ArcRequestItem Class Reference

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

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

5.12.1 Detailed Description

Container, <Subjects, Actions, Objects, Contexts> tuple. Specified **ArcRequestItem** (p. 21) which can parse Arc request format

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

- ArcRequestItem.h

5.13 ArcSec::ArcRequestTuple Class Reference

RequestTuple, container which includes the.

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

5.13.1 Detailed Description

RequestTuple, container which includes the.

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

- ArcEvaluationCtx.h

5.14 ArcSec::ArcRule Class Reference

ArcRule (p. 21) class to parse Arc specific <Rule> node.

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

5.14.1 Detailed Description

ArcRule (p. 21) class to parse Arc specific <Rule> node.

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

- ArcRule.h

5.15 ArcSec::AttributeDesignator Class Reference

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

- AttributeDesignator.h

5.16 ArcSec::AttributeSelector Class Reference

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

- AttributeSelector.h

5.17 Arc::ConfigTLMCC Class Reference

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

- ConfigTLMCC.h

5.18 Arc::DataPointARC Class Reference

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

- DataPointARC.h

5.19 Arc::DataPointFile Class Reference

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

- DataPointFile.h

5.20 Arc::DataPointGridFTP Class Reference

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

- DataPointGridFTP.h

5.21 Arc::DataPointHTTP Class Reference

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

- DataPointHTTP.h

5.22 Arc::DataPointLDAP Class Reference

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

- DataPointLDAP.h

5.23 Arc::DataPointLFC Class Reference

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

- DataPointLFC.h

5.24 Arc::DataPointRLS Class Reference

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

- DataPointRLS.h

5.25 Arc::DataPointSRM Class Reference

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

- DataPointSRM.h

5.26 ArcSec::DelegationCollector Class Reference

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

- DelegationCollector.h

5.27 ArcSec::DelegationMultiSecAttr Class Reference

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

- DelegationSecAttr.h

5.28 ArcSec::DelegationPDP Class Reference

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

5.28.1 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

5.29 ArcSec::DelegationSecAttr Class Reference

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

- DelegationSecAttr.h

5.30 ArcSec::DelegationSH Class Reference

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

- DelegationSH.h

5.31 ArcSec::DenyPDP Class Reference

This PDP always returns false (deny).

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

5.31.1 Detailed Description

This PDP always returns false (deny).

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

- DenyPDP.h

5.32 ArcSec::GACLEvaluator Class Reference

Public Member Functions

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

5.32.1 Member Function Documentation

5.32.1.1 virtual Response* ArcSec::GACLEvaluator::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:

- GACLEvaluator.h

5.33 ArcSec::GACLPDP Class Reference

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

- GACLPDP.h

5.34 ArcSec::GACLPolicy Class Reference

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

- GACLPolicy.h

5.35 ArcSec::GACLRequest Class Reference

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

- GACLRequest.h

5.36 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)

5.36.1 Detailed Description

LDAPQuery (p. 25) class; querying of LDAP servers.

5.36.2 Constructor & Destructor Documentation

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

Constructs a new **LDAPQuery** (p. 25) object and sets connection options. The connection is first established when calling **Query**.

5.36.2.2 Arc::LDAPQuery::~~LDAPQuery ()

Destructor. Will disconnect from the ldapserver if still connected.

5.36.3 Member Function Documentation

5.36.3.1 `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.

5.36.3.2 `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

5.37 Arc::Lister Class Reference

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

- Lister.h

5.38 Arc::MCC_GSI_Client Class Reference

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

- MCCGSI.h

5.39 Arc::MCC_GSI_Service Class Reference

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

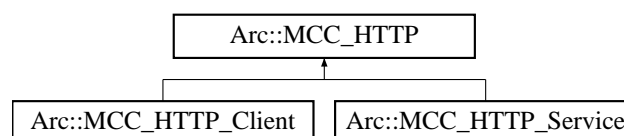
- MCCGSI.h

5.40 Arc::MCC_HTTP Class Reference

A base class for HTTP client and service MCCs.

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

Inheritance diagram for Arc::MCC_HTTP:



5.40.1 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.

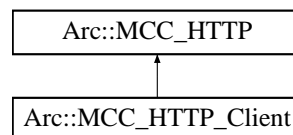
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.41 Arc::MCC_HTTP_Client Class Reference

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

Inheritance diagram for Arc::MCC_HTTP_Client:



5.41.1 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 method 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.

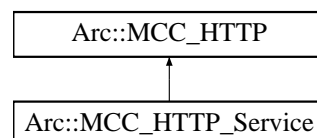
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.42 Arc::MCC_HTTP_Service Class Reference

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

Inheritance diagram for Arc::MCC_HTTP_Service:



5.42.1 Detailed Description

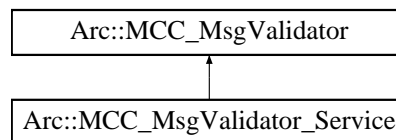
This class implements MCC to processes HTTP request. On input payload with PayloadStreamInterface is expected. HTTP message is read from stream and its body is converted into PayloadRaw and passed to next MCC. Returned payload of PayloadRawInterface type is treated as body part of returning **PayloadHTTP** (p. 33). 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

5.43 Arc::MCC_MsgValidator Class Reference

Inheritance diagram for Arc::MCC_MsgValidator:

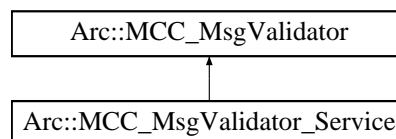


The documentation for this class was generated from the following file:

- MCCMsgValidator.h

5.44 Arc::MCC_MsgValidator_Service Class Reference

Inheritance diagram for Arc::MCC_MsgValidator_Service:



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

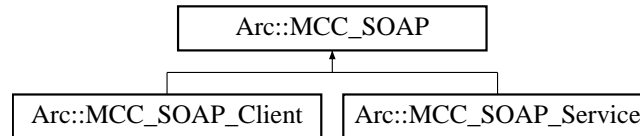
- MCCMsgValidator.h

5.45 Arc::MCC_SOAP Class Reference

A base class for SOAP client and service MCCs.

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

Inheritance diagram for Arc::MCC_SOAP:



5.45.1 Detailed Description

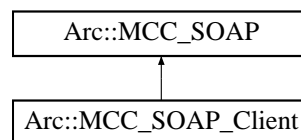
A base class for SOAP client and service MCCs. This is a base class for SOAP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

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

- MCCSOAP.h

5.46 Arc::MCC_SOAP_Client Class Reference

Inheritance diagram for Arc::MCC_SOAP_Client:



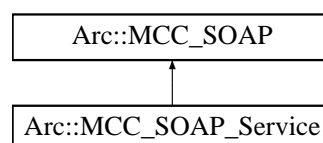
The documentation for this class was generated from the following file:

- MCCSOAP.h

5.47 Arc::MCC_SOAP_Service Class Reference

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

Inheritance diagram for Arc::MCC_SOAP_Service:



5.47.1 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:

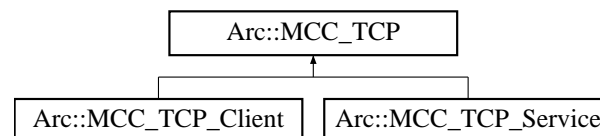
- MCCSOAP.h

5.48 Arc::MCC_TCP Class Reference

A base class for TCP client and service MCCs.

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

Inheritance diagram for Arc::MCC_TCP:



5.48.1 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.

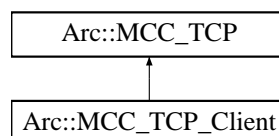
The documentation for this class was generated from the following file:

- MCCTCP.h

5.49 Arc::MCC_TCP_Client Class Reference

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

Inheritance diagram for Arc::MCC_TCP_Client:



5.49.1 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.

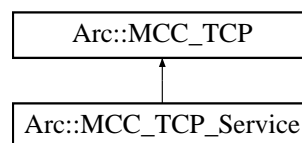
The documentation for this class was generated from the following file:

- MCCTCP.h

5.50 Arc::MCC_TCP_Service Class Reference

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

Inheritance diagram for Arc::MCC_TCP_Service:



Data Structures

- class `mcc_tcp_exec_t`
- class `mcc_tcp_handle_t`

Public Member Functions

- `MCC_TCP_Service` (Config *cfg)

5.50.1 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`

5.50.2 Constructor & Destructor Documentation

5.50.2.1 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:

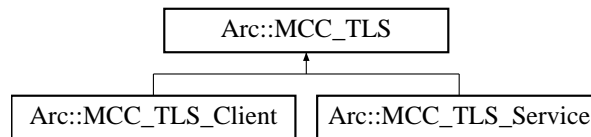
- MCCTCP.h

5.51 Arc::MCC_TLS Class Reference

A base class for TLS client and service MCCs.

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

Inheritance diagram for Arc::MCC_TLS:



5.51.1 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.

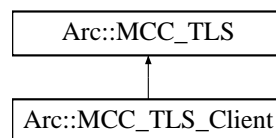
The documentation for this class was generated from the following file:

- MCCTLS.h

5.52 Arc::MCC_TLS_Client Class Reference

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

Inheritance diagram for Arc::MCC_TLS_Client:



5.52.1 Detailed Description

This class is MCC implementing TLS client.

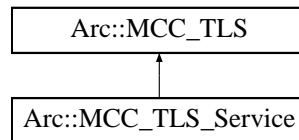
The documentation for this class was generated from the following file:

- MCCTLS.h

5.53 Arc::MCC_TLS_Service Class Reference

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

Inheritance diagram for Arc::MCC_TLS_Service:



5.53.1 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. 30), 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. 27) calls the process() method of **MCC_TLS_Client** (p. 32) object, or **MCC_TCP_Service** (p. 31) calls the process() method of **MCC_TLS_Service** (p. 32) object. The "ssl" object is embeded in a payload called PayloadTLSSocket.

The process() method of **MCC_TLS_Service** (p. 32) is passed payload implementing PayloadStreamInterface 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 remote peer.

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

- MCCTLS.h

5.54 Arc::PayloadGSISStream Class Reference

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

- PayloadGSISStream.h

5.55 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_**

5.55.1 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.

5.55.2 Constructor & Destructor Documentation

5.55.2.1 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.

5.55.2.2 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.

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

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

5.55.2.4 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.

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

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

5.55.3 Member Function Documentation

5.55.3.1 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.

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

Adds HTTP header attribute 'name' = 'value'

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

Returns all HTTP header attributes.

5.55.3.4 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.

5.55.3.5 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.

5.55.3.6 bool Arc::PayloadHTTP::get_body (void) [protected]

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

5.55.3.7 bool Arc::PayloadHTTP::parse_header (void) [protected]

Read HTTP header and fill internal variables

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

Read up to 'size' bytes from stream_

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

Read from stream till

5.55.4 Field Documentation

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

true if connection should not be closed after response

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

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

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

Content-length of HTTP message

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

HTTP method being used or requested

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

true if content is chunked

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

HTTP reason being sent or supplied

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

minor number of HTTP version - must be 0 or 1

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

if true stream_ is owned by this

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

HTTP code being sent or supplied

5.55.4.10 PayloadStreamInterface* Arc::PayloadHTTP::sbody_ [protected]

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

5.55.4.11 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_.

5.55.4.12 bool Arc::PayloadHTTP::stream_own_ [protected]

stream used to communicate to outside

5.55.4.13 std::string Arc::PayloadHTTP::uri_ [protected]

if true body_ is owned by this

5.55.4.14 int Arc::PayloadHTTP::version_major_ [protected]

URI being contacted

5.55.4.15 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

5.56 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)

5.56.1 Detailed Description

This class extends PayloadStream with TCP socket specific features

5.56.2 Constructor & Destructor Documentation

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

Constructor - connects to TCP server at specified hostname:port

5.56.2.2 `Arc::PayloadTCPSocket::PayloadTCPSocket (const std::string endpoint, int timeout, Logger & logger)`

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

5.56.2.3 `Arc::PayloadTCPSocket::PayloadTCPSocket (int s, int timeout, Logger & logger) [inline]`

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

5.56.2.4 `Arc::PayloadTCPSocket::PayloadTCPSocket (PayloadTCPSocket & s) [inline]`

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

5.56.2.5 `Arc::PayloadTCPSocket::PayloadTCPSocket (PayloadTCPSocket & s, Logger & logger) [inline]`

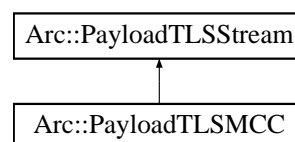
Copy constructor - inherits handle of copied object. Handle is NOT closed in destructor.

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

- PayloadTCPSocket.h

5.57 Arc::PayloadTLMCC Class Reference

Inheritance diagram for Arc::PayloadTLMCC:



Public Member Functions

- `PayloadTLMCC` (MCCInterface *mcc, const **ConfigTLMCC** &cfg, Logger &logger)
- `PayloadTLMCC` (PayloadStreamInterface *stream, const **ConfigTLMCC** &cfg, Logger &logger)
- `PayloadTLMCC` (PayloadTLMCC &stream)

5.57.1 Constructor & Destructor Documentation

5.57.1.1 Arc::PayloadTLMCC::PayloadTLMCC (MCCInterface * *mcc*, const ConfigTLMCC & *cfg*, Logger & *logger*)

Constructor - creates ssl object which is bound to next MCC. This instance must be used on client side. It obtains Stream interface from next MCC dynamically.

5.57.1.2 Arc::PayloadTLMCC::PayloadTLMCC (PayloadStreamInterface * *stream*, const ConfigTLMCC & *cfg*, Logger & *logger*)

Constructor - creates ssl object which is bound to stream. This constructor to be used on server side. Provided stream is NOT destroyed in destructor.

5.57.1.3 Arc::PayloadTLMCC::PayloadTLMCC (PayloadTLMCC & *stream*)

Copy constructor with new logger. Created object shares same SSL objects but does not destroy them in destructor. Main instance must be destroyed after all copied ones.

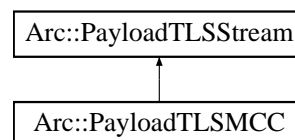
The documentation for this class was generated from the following file:

- PayloadTLMCC.h

5.58 Arc::PayloadTLSStream Class Reference

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

Inheritance diagram for Arc::PayloadTLSStream:



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_**

5.58.1 Detailed Description

Implementation of PayloadStreamInterface for SSL handle.

5.58.2 Constructor & Destructor Documentation

5.58.2.1 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.

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

Destructor.

5.58.3 Member Function Documentation

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

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

5.58.3.2 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.

5.58.3.3 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.

5.58.4 Field Documentation

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

Timeout for read/write operations

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

- PayloadTLSStream.h

5.59 ArcSec::PDPSERVICEInvoker Class Reference

PDPSERVICEInvoker (p. 40) - client which will invoke pdpservice.

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

5.59.1 Detailed Description

PDPServiceInvoker (p. 40) - client which will invoke pdpservice.

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

- PDPServiceInvoker.h

5.60 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference

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

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

5.60.1 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

5.61 ArcSec::SAMLTokenSH Class Reference

Adds WS-Security SAML Token into SOAP Header.

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

5.61.1 Detailed Description

Adds WS-Security SAML Token into SOAP Header.

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

- SAMLTokenSH.h

5.62 ArcSec::SimpleListPDP Class Reference

Tests X509 subject against list of subjects in file.

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

5.62.1 Detailed Description

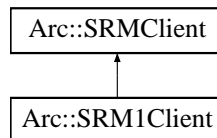
Tests X509 subject against list of subjects in file. This class implements PDP interface. It's isPermitted() method compares X509 subject of requestor obtained from TLS layer (TLS:PEERDN) to list of subjects (one per line) in external file. Location of file is defined by 'location' attribute of PDP configuration.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

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

- SimpleListPDP.h

5.63 Arc::SRM1Client Class Reference

Inheritance diagram for Arc::SRM1Client:



Public Member Functions

- SRMReturnCode **ping** (std::string &, bool=true)
- SRMReturnCode **getSpaceTokens** (std::list< std::string > &, const std::string &="")
- SRMReturnCode **getRequestTokens** (std::list< std::string > &, const std::string &="")
- SRMReturnCode **requestBringOnline** (SRMClientRequest &)
- SRMReturnCode **requestBringOnlineStatus** (SRMClientRequest &)
- SRMReturnCode **mkDir** (SRMClientRequest &)
- SRMReturnCode **getTURLs** (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode **putTURLs** (SRMClientRequest &req, std::list< std::string > &urls, const unsigned long long size=0)
- SRMReturnCode **releaseGet** (SRMClientRequest &req)
- SRMReturnCode **releasePut** (SRMClientRequest &req)
- SRMReturnCode **release** (SRMClientRequest &req)
- SRMReturnCode **abort** (SRMClientRequest &req)
- SRMReturnCode **info** (SRMClientRequest &req, std::list< struct SRMFileMetaData > &meta-data, const int recursive=0, bool report_error=true)
- SRMReturnCode **remove** (SRMClientRequest &req)
- SRMReturnCode **copy** (SRMClientRequest &req, const std::string &source)

5.63.1 Member Function Documentation

5.63.1.1 SRMReturnCode Arc::SRM1Client::abort (SRMClientRequest & *req*) [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

Implements **Arc::SRMClient** (p. 52).

5.63.1.2 SRMReturnCode Arc::SRM1Client::copy (SRMClientRequest & *req*, const std::string & *source*) [virtual]

Copy a file between two SRM storages.

Parameters

req The request object

source The source URL

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 52).

5.63.1.3 SRMReturnCode Arc::SRM1Client::getRequestTokens (std::list< std::string > & *tokens*, const std::string & *description* = "") [inline, 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

Implements **Arc::SRMClient** (p. 53).

5.63.1.4 SRMReturnCode Arc::SRM1Client::getSpaceTokens (std::list< std::string > & *tokens*, const std::string & *description* = "") [inline, 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

Implements **Arc::SRMClient** (p. 53).

5.63.1.5 SRMReturnCode Arc::SRM1Client::getURLs (SRMClientRequest & *req*, std::list< std::string > & *urls*) [virtual]

If the user wishes to copy a file from somewhere, **getURLs()** (p. 44) is called to retrieve the transport URL to copy the file from.

Parameters

req The request object
urls A list of URLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 53).

5.63.1.6 SRMReturnCode Arc::SRM1Client::info (SRMClientRequest & *req*, std::list< struct SRMFileMetaData > & *metadata*, const int *recursive* = 0, bool *report_error* = true) [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. 60)

Implements **Arc::SRMClient** (p. 54).

5.63.1.7 SRMReturnCode Arc::SRM1Client::mkDir (SRMClientRequest & *req*) [inline, virtual]

Make required directories for the SURL in the request

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 54).

5.63.1.8 SRMReturnCode Arc::SRM1Client::ping (std::string & *version*, bool *report_error* = *true*) [inline, 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 supresses 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

Implements **Arc::SRMClient** (p. 54).

5.63.1.9 SRMReturnCode Arc::SRM1Client::putTURLs (SRMClientRequest & *req*, std::list< std::string > & *urls*, const unsigned long long *size* = 0) [virtual]

If the user wishes to copy a file to somewhere, **putTURLs()** (p. 45) is called to retrieve the transport URL to copy the file to.

Parameters

req The request object
urls A list of TURLs filled by the method
size The size of the file

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 55).

5.63.1.10 SRMReturnCode Arc::SRM1Client::release (SRMClientRequest & *req*) [virtual]

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

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 55).

5.63.1.11 SRMReturnCode Arc::SRM1Client::releaseGet (SRMClientRequest & *req*) [virtual]

Should be called after a successful copy from SRM storage.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 55).

5.63.1.12 SRMReturnCode Arc::SRM1Client::releasePut (SRMClientRequest & *req*) [virtual]

Should be called after a successful copy to SRM storage.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 56).

5.63.1.13 SRMReturnCode Arc::SRM1Client::remove (SRMClientRequest & *req*) [virtual]

Delete a file physically from storage and the SRM namespace.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 56).

5.63.1.14 SRMReturnCode Arc::SRM1Client::requestBringOnline (SRMClientRequest & *req*) [inline, virtual]

Submit a request to bring online files. This operation is asynchronous and the status of the request can be checked by calling **requestBringOnlineStatus()** (p. 47) with the request token in *req* which is assigned by this method.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **Arc::SRMClient** (p. 56).

5.63.1.15 SRMReturnCode Arc::SRM1Client::requestBringOnlineStatus (SRMClientRequest & req) [inline, virtual]

Query the status of a request to bring files online. The SURLs map is updated if the status of any files in the request has changed.

Parameters

req The request object to query the status of

Returns

SRMReturnCode specifying outcome of operation

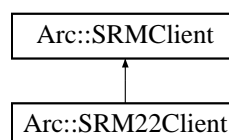
Implements **Arc::SRMClient** (p. 57).

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

- SRM1Client.h

5.64 Arc::SRM22Client Class Reference

Inheritance diagram for Arc::SRM22Client:

**Public Member Functions**

- **SRM22Client** (const UserConfig &usercfg, const **SRMURL** &url)
- **~SRM22Client** ()
- SRMReturnCode **ping** (std::string &version, bool report_error=true)
- SRMReturnCode **getSpaceTokens** (std::list< std::string > &tokens, const std::string &description="")
- SRMReturnCode **getRequestTokens** (std::list< std::string > &tokens, const std::string &description="")
- SRMReturnCode **getTURLs** (**SRMClientRequest** &req, std::list< std::string > &urls)
- SRMReturnCode **putTURLs** (**SRMClientRequest** &req, std::list< std::string > &urls, const unsigned long long size=0)
- SRMReturnCode **requestBringOnline** (**SRMClientRequest** &req)
- SRMReturnCode **requestBringOnlineStatus** (**SRMClientRequest** &req)

- SRMReturnCode **info** (SRMClientRequest &req, std::list< struct SRMFileMetaData > &meta-data, const int recursive=0, bool report_error=true)
- SRMReturnCode **releaseGet** (SRMClientRequest &req)
- SRMReturnCode **releasePut** (SRMClientRequest &req)
- SRMReturnCode **release** (SRMClientRequest &)
- SRMReturnCode **abort** (SRMClientRequest &req)
- SRMReturnCode **remove** (SRMClientRequest &req)
- SRMReturnCode **copy** (SRMClientRequest &req, const std::string &source)
- SRMReturnCode **mkDir** (SRMClientRequest &req)

5.64.1 Constructor & Destructor Documentation

5.64.1.1 Arc::SRM22Client::SRM22Client (const UserConfig & *usercfg*, const SRMURL & *url*)

Constructor

5.64.1.2 Arc::SRM22Client::~~SRM22Client ()

Destructor

5.64.2 Member Function Documentation

5.64.2.1 SRMReturnCode Arc::SRM22Client::abort (SRMClientRequest & *req*) [virtual]

Abort request. Called after any failure in the data transfer or putDone calls

Implements Arc::SRMClient (p. 52).

5.64.2.2 SRMReturnCode Arc::SRM22Client::copy (SRMClientRequest & *req*, const std::string & *source*) [virtual]

Implemented in pull mode, ie the endpoint defined in the request object performs the copy.

Implements Arc::SRMClient (p. 52).

5.64.2.3 SRMReturnCode Arc::SRM22Client::getRequestTokens (std::list< std::string > & *tokens*, const std::string & *description* = "") [virtual]

Use srmGetRequestTokens to return a list of spaces available

Implements Arc::SRMClient (p. 53).

5.64.2.4 SRMReturnCode Arc::SRM22Client::getSpaceTokens (std::list< std::string > & *tokens*, const std::string & *description* = "") [virtual]

Use srmGetSpaceTokens to return a list of spaces available

Implements Arc::SRMClient (p. 53).

5.64.2.5 SRMReturnCode Arc::SRM22Client::getTURLs (SRMClientRequest & req, std::list< std::string > & urls) [virtual]

Get a list of TURLs for the given SURL. Uses srmPrepareToGet and waits until file is ready (online and pinned). Although a list is returned, SRMv2.2 only returns one TURL per SURL.

Implements **Arc::SRMClient** (p. 53).

5.64.2.6 SRMReturnCode Arc::SRM22Client::info (SRMClientRequest & req, std::list< struct SRMFileMetaData > & metadata, const int recursive = 0, bool report_error = true) [virtual]

Use srmLs to get info on the given SURL. Info on each file is put in a metadata struct and added to the list.

Implements **Arc::SRMClient** (p. 54).

5.64.2.7 SRMReturnCode Arc::SRM22Client::mkDir (SRMClientRequest & req) [virtual]

Call srmMkDir

Implements **Arc::SRMClient** (p. 54).

5.64.2.8 SRMReturnCode Arc::SRM22Client::ping (std::string & version, bool report_error = true) [virtual]

Get the server version from srmPing

Implements **Arc::SRMClient** (p. 54).

5.64.2.9 SRMReturnCode Arc::SRM22Client::putTURLs (SRMClientRequest & req, std::list< std::string > & urls, const unsigned long long size = 0) [virtual]

Retrieve TURLs which a file can be written to. Uses srmPrepareToPut and waits until a suitable TURL has been assigned. Although a list is returned, SRMv2.2 only returns one TURL per SURL.

Implements **Arc::SRMClient** (p. 55).

5.64.2.10 SRMReturnCode Arc::SRM22Client::release (SRMClientRequest &) [inline, virtual]

Not used in this version of SRM

Implements **Arc::SRMClient** (p. 55).

5.64.2.11 SRMReturnCode Arc::SRM22Client::releaseGet (SRMClientRequest & req) [virtual]

Release files that have been pinned by srmPrepareToGet using srmReleaseFiles. Called after successful file transfer or failed prepareToGet.

Implements **Arc::SRMClient** (p. 55).

5.64.2.12 SRMReturnCode Arc::SRM22Client::releasePut (SRMClientRequest & *req*) [virtual]

Mark a put request as finished. Called after successful file transfer or failed prepareToPut.

Implements **Arc::SRMClient** (p. 56).

5.64.2.13 SRMReturnCode Arc::SRM22Client::remove (SRMClientRequest & *req*) [virtual]

Delete by srmRm or srmRmDir

Implements **Arc::SRMClient** (p. 56).

5.64.2.14 SRMReturnCode Arc::SRM22Client::requestBringOnline (SRMClientRequest & *req*) [virtual]

Call srmBringOnline with the URLs specified in req.

Implements **Arc::SRMClient** (p. 56).

5.64.2.15 SRMReturnCode Arc::SRM22Client::requestBringOnlineStatus (SRMClientRequest & *req*) [virtual]

Call srmStatusOfBringOnlineRequest and update req with any changes.

Implements **Arc::SRMClient** (p. 57).

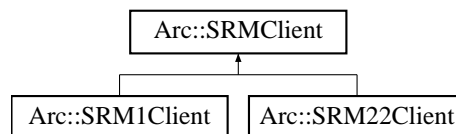
The documentation for this class was generated from the following file:

- SRM22Client.h

5.65 Arc::SRMClient Class Reference

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

Inheritance diagram for Arc::SRMClient:



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 **getURLs** (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode **requestBringOnline** (SRMClientRequest &req)=0
- virtual SRMReturnCode **requestBringOnlineStatus** (SRMClientRequest &req)=0
- virtual SRMReturnCode **putURLs** (SRMClientRequest &req, std::list< std::string > &urls, const unsigned long long size=0)=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 &timed-out, time_t timeout=300)
- static void **Timeout** (const time_t t)

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 **request_timeout**
- static Logger **logger**

5.65.1 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.

5.65.2 Constructor & Destructor Documentation

5.65.2.1 `Arc::SRMClient::SRMClient (const UserConfig & usercfg, const SRMURL & url) [protected]`

Constructor

5.65.2.2 `virtual Arc::SRMClient::~~SRMClient () [virtual]`

Destructor

5.65.3 Member Function Documentation

5.65.3.1 `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

Implemented in `Arc::SRM1Client` (p.42), and `Arc::SRM22Client` (p.48).

5.65.3.2 `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 URL

Returns

SRMReturnCode specifying outcome of operation

Implemented in `Arc::SRM1Client` (p.43), and `Arc::SRM22Client` (p.48).

5.65.3.3 `static SRMClient* Arc::SRMClient::getInstance (const UserConfig & usercfg, const std::string & url, bool & timedout, time_t timeout = 300) [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

timeout Connection timeout. is returned.

5.65.3.4 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

Implemented in **Arc::SRM1Client** (p. 43), and **Arc::SRM22Client** (p. 48).

5.65.3.5 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

Implemented in **Arc::SRM1Client** (p. 43), and **Arc::SRM22Client** (p. 48).

5.65.3.6 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 to copy the file from.

Parameters

req The request object

urls A list of TURLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

Implemented in **Arc::SRM1Client** (p. 44), and **Arc::SRM22Client** (p. 49).

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

Returns the version of the SRM protocol used by this instance

References version.

5.65.3.8 `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. 60)

Implemented in [Arc::SRM1Client](#) (p. 44), and [Arc::SRM22Client](#) (p. 49).

5.65.3.9 `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

Implemented in [Arc::SRM1Client](#) (p. 44), and [Arc::SRM22Client](#) (p. 49).

5.65.3.10 `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

Implemented in **Arc::SRM1Client** (p. 45), and **Arc::SRM22Client** (p. 49).

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

Process SOAP request

5.65.3.12 virtual SRMReturnCode Arc::SRMClient::putURLs (SRMClientRequest & *req*, std::list< std::string > & *urls*, const unsigned long long *size* = 0) [pure virtual]

If the user wishes to copy a file to somewhere, **putURLs()** (p. 55) is called to retrieve the transport URL to copy the file to.

Parameters

req The request object

urls A list of TURLs filled by the method

size The size of the file

Returns

SRMReturnCode specifying outcome of operation

Implemented in **Arc::SRM1Client** (p. 45), and **Arc::SRM22Client** (p. 49).

5.65.3.13 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

Implemented in **Arc::SRM1Client** (p. 45), and **Arc::SRM22Client** (p. 49).

5.65.3.14 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

Implemented in **Arc::SRM1Client** (p. 46), and **Arc::SRM22Client** (p. 49).

5.65.3.15 **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

Implemented in **Arc::SRM1Client** (p. 46), and **Arc::SRM22Client** (p. 50).

5.65.3.16 **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

Implemented in **Arc::SRM1Client** (p. 46), and **Arc::SRM22Client** (p. 50).

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

Submit a request to bring online files. 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.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **Arc::SRM1Client** (p. 46), and **Arc::SRM22Client** (p. 50).

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

Query the status of a request to bring files online. The URLs map is updated if the status of any files in the request has changed.

Parameters

req The request object to query the status of

Returns

SRMReturnCode specifying outcome of operation

Implemented in **Arc::SRM1Client** (p. 47), and **Arc::SRM22Client** (p. 50).

5.65.3.19 static void Arc::SRMClient::Timeout (const time_t *t*) [inline, static]

set the request timeout

References request_timeout.

5.65.4 Field Documentation**5.65.4.1 MCCCConfig Arc::SRMClient::cfg [protected]**

SOAP configuraton object

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

SOAP client object

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

The implementation of the server

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

Logger

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

SOAP namespace

5.65.4.6 time_t Arc::SRMClient::request_timeout [static, protected]

Timeout for requests to the SRM service

Referenced by Timeout().

5.65.4.7 `std::string Arc::SRMClient::service_endpoint` `[protected]`

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

5.65.4.8 `time_t Arc::SRMClient::user_timeout` `[protected]`

Timeout for requests to the SRM service

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

The version of the SRM protocol used

Referenced by `getVersion()`.

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

- `SRMClient.h`

5.66 `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 **`long_list`** (bool list)

5.66.1 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()`.

5.66.2 Constructor & Destructor Documentation

5.66.2.1 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

5.66.2.2 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

5.66.3 Member Function Documentation

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

set and get file id list

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

set and get status of request

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

set and get long list flag

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

set and get request id

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

set and get request token

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

set and get space token

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

set and get surl failures

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

set and get surl statuses

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

get URLs

5.66.3.10 `void Arc::SRMClientRequest::waiting_time (int wait_time) [inline]`

set and get waiting time

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

- SRMClient.h

5.67 SRMFileInfo Class Reference

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

5.67.1 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

5.68 Arc::SRMFileMetaData Struct Reference

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

5.68.1 Detailed Description

File metadata

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

- SRMClient.h

5.69 SRMInfo Class Reference

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

5.69.1 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

5.70 Arc::SRMInvalidRequestException Class Reference

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

- SRMClient.h

5.71 SRMURL Class Reference

Public Member Functions

- **SRMURL** (std::string url)
- const std::string & **Endpoint** (void) const
- void **SetSRMVersion** (const std::string &version)
- const std::string & **FileName** (void) const
- std::string **ContactURL** (void) const
- std::string **BaseURL** (void) const
- std::string **ShortURL** (void) const
- std::string **FullURL** (void) const
- bool **PortDefined** ()

5.71.1 Constructor & Destructor Documentation

5.71.1.1 SRMURL::SRMURL (std::string url)

Examples shown for functions below assume the object was initiated with srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/dummyfile3

5.71.2 Member Function Documentation

5.71.2.1 std::string SRMURL::BaseURL (void) const

eg srm://srm.ndgf.org:8443/srm/managerv2?SFN=

5.71.2.2 std::string SRMURL::ContactURL (void) const

eg http://srm.ndgf.org:8443/srm/managerv2

5.71.2.3 const std::string& SRMURL::Endpoint (void) const [inline]

eg /srm/managerv2

5.71.2.4 `const std::string& SRMURL::FileName (void) const [inline]`

eg pnfs/ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/dummyfile3

5.71.2.5 `std::string SRMURL::FullURL (void) const`

eg srm://srm.ndgf.org:8443/srm/managerv2?SFN=pnfs/ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/dummyfile3

5.71.2.6 `bool SRMURL::PortDefined () [inline]`

Was the port number given in the constructor?

5.71.2.7 `void SRMURL::SetSRMVersion (const std::string & version)`

Possible values of version are "1" and "2.2"

5.71.2.8 `std::string SRMURL::ShortURL (void) const`

eg srm://srm.ndgf.org:8443/pnfs/ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/dummyfile3

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

- SRMURL.h

5.72 ArcSec::UsernameTokenSH Class Reference

Adds WS-Security Username Token into SOAP Header.

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

5.72.1 Detailed Description

Adds WS-Security Username Token into SOAP Header.

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

- UsernameTokenSH.h

5.73 ArcSec::X509TokenSH Class Reference

Adds WS-Security X509 Token into SOAP Header.

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

5.73.1 Detailed Description

Adds WS-Security X509 Token into SOAP Header.

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

- X509TokenSH.h

5.74 ArcSec::XACMLAlgFactory Class Reference

Algorithm factory class for XACML.

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

Public Member Functions

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

5.74.1 Detailed Description

Algorithm factory class for XACML.

5.74.2 Member Function Documentation

5.74.2.1 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. 63) itself will release the Alg objects

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

- XACMLAlgFactory.h

5.75 ArcSec::XACMLApply Class Reference

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

- XACMLApply.h

5.76 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)

5.76.1 Detailed Description

Attribute factory class for XACML specified attributes.

5.76.2 Member Function Documentation

5.76.2.1 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

5.77 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference

XACML specific AttributeProxy class.

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

Public Member Functions

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

5.77.1 Detailed Description

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

XACML specific AttributeProxy class.

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

- XACMLAttributeProxy.h

5.78 ArcSec::XACMLCondition Class Reference

XACMLCondition (p. 64) class to parse and operate XACML specific <Condition> node.

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

Public Member Functions

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

5.78.1 Detailed Description

XACMLCondition (p. 64) class to parse and operate XACML specific <Condition> node.

5.78.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLCondition.h

5.79 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)

5.79.1 Detailed Description

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

5.79.2 Constructor & Destructor Documentation

5.79.2.1 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

5.80 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)

5.80.1 Detailed Description

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

5.80.2 Member Function Documentation

5.80.2.1 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

5.81 ArcSec::XACMLFnFactory Class Reference

Function factory class for XACML specified attributes.

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

Public Member Functions

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

5.81.1 Detailed Description

Function factory class for XACML specified attributes.

5.81.2 Member Function Documentation

5.81.2.1 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. 66) itself will release the Function objects

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

- XACMLFnFactory.h

5.82 ArcSec::XACMLPDP Class Reference

XACMLPDP (p. 67) - PDP which can handle the XACML specific request and policy schema.

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

5.82.1 Detailed Description

XACMLPDP (p. 67) - PDP which can handle the XACML specific request and policy schema.

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

- XACMLPDP.h

5.83 ArcSec::XACMLPolicy Class Reference

XACMLPolicy (p. 67) 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** ()

5.83.1 Detailed Description

XACMLPolicy (p. 67) class to parse and operate XACML specific <Policy> node.

5.83.2 Constructor & Destructor Documentation

5.83.2.1 ArcSec::XACMLPolicy::XACMLPolicy (void)

Constructor

5.83.2.2 ArcSec::XACMLPolicy::XACMLPolicy (const Arc::XMLNode *node*)

Constructor

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

Constructor -

5.83.3 Member Function Documentation

5.83.3.1 `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

5.84 ArcSec::XACMLRequest Class Reference

Public Member Functions

- `virtual const char * getEvalName () const`
- `virtual const char * getName () const`

5.84.1 Member Function Documentation

5.84.1.1 `virtual const char* ArcSec::XACMLRequest::getEvalName () const [inline, virtual]`

Get the name of corresponding evaluator

5.84.1.2 `virtual const char* ArcSec::XACMLRequest::getName (void) const [inline, virtual]`

Get the name of this request

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

- XACMLRequest.h

5.85 ArcSec::XACMLRule Class Reference

XACMLRule (p. 68) class to parse XACML specific <Rule> node.

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

5.85.1 Detailed Description

XACMLRule (p. 68) class to parse XACML specific <Rule> node.

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

- XACMLRule.h

5.86 ArcSec::XACMLTarget Class Reference

XACMLTarget (p. 69) class to parse and operate XACML specific <Target> node.

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

Public Member Functions

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

5.86.1 Detailed Description

XACMLTarget (p. 69) class to parse and operate XACML specific <Target> node.

5.86.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLTarget.h

5.87 ArcSec::XACMLTargetMatch Class Reference

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

- XACMLTarget.h

5.88 ArcSec::XACMLTargetMatchGroup Class Reference

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

- XACMLTarget.h

5.89 ArcSec::XACMLTargetSection Class Reference

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

- XACMLTarget.h

Index

- ~LDAPQuery
 - Arc::LDAPQuery, 25
- ~PayloadTLSStream
 - Arc::PayloadTLSStream, 40
- ~SRM22Client
 - Arc::SRM22Client, 48
- ~SRMClient
 - Arc::SRMClient, 52
- abort
 - Arc::SRM1Client, 42
 - Arc::SRM22Client, 48
 - Arc::SRMClient, 52
- AndList
 - ArcSec, 14
- Arc::ConfigTLSMCC, 22
- Arc::DataPointARC, 22
- Arc::DataPointFile, 22
- Arc::DataPointGridFTP, 22
- Arc::DataPointHTTP, 22
- Arc::DataPointLDAP, 22
- Arc::DataPointLFC, 23
- Arc::DataPointRLS, 23
- Arc::DataPointSRM, 23
- Arc::LDAPQuery, 25
 - ~LDAPQuery, 25
 - LDAPQuery, 25
 - Query, 26
 - Result, 26
- Arc::Lister, 26
- Arc::MCC_GSI_Client, 26
- Arc::MCC_GSI_Service, 26
- Arc::MCC_HTTP, 26
- Arc::MCC_HTTP_Client, 27
- Arc::MCC_HTTP_Service, 27
- Arc::MCC_MsgValidator, 28
- Arc::MCC_MsgValidator_Service, 28
- Arc::MCC_SOAP, 28
- Arc::MCC_SOAP_Client, 29
- Arc::MCC_SOAP_Service, 29
- Arc::MCC_TCP, 30
- Arc::MCC_TCP_Client, 30
- Arc::MCC_TCP_Service, 31
 - MCC_TCP_Service, 31
- Arc::MCC_TLS, 32
- Arc::MCC_TLS_Client, 32
- Arc::MCC_TLS_Service, 32
- Arc::PayloadGSISStream, 33
- Arc::PayloadHTTP, 33
 - Attribute, 35
 - Attributes, 35
 - attributes_, 36
 - Body, 35
 - body_own_, 36
 - chunked_, 36
 - code_, 36
 - Flush, 35
 - get_body, 35
 - keep_alive_, 36
 - length_, 36
 - method_, 36
 - parse_header, 35
 - PayloadHTTP, 34, 35
 - rbody_, 36
 - read, 35
 - readline, 36
 - reason_, 36
 - sbody_, 36
 - stream_, 37
 - stream_own_, 37
 - uri_, 37
 - version_major_, 37
 - version_minor_, 37
- Arc::PayloadTCPSocket, 37
 - PayloadTCPSocket, 38
- Arc::PayloadTLSMCC, 38
 - PayloadTLSMCC, 39
- Arc::PayloadTLSStream, 39
 - ~PayloadTLSStream, 40
 - GetCert, 40
 - GetPeerCert, 40
 - PayloadTLSStream, 40
 - ssl_, 40
 - STACK_OF, 40
- Arc::SRM1Client, 42
 - abort, 42
 - copy, 42
 - getRequestTokens, 43
 - getSpaceTokens, 43
 - getURLs, 43

- info, 44
- mkDir, 44
- ping, 44
- putURLs, 45
- release, 45
- releaseGet, 45
- releasePut, 46
- remove, 46
- requestBringOnline, 46
- requestBringOnlineStatus, 47
- Arc::SRM22Client, 47
 - ~SRM22Client, 48
 - abort, 48
 - copy, 48
 - getRequestTokens, 48
 - getSpaceTokens, 48
 - getURLs, 48
 - info, 49
 - mkDir, 49
 - ping, 49
 - putURLs, 49
 - release, 49
 - releaseGet, 49
 - releasePut, 49
 - remove, 50
 - requestBringOnline, 50
 - requestBringOnlineStatus, 50
 - SRM22Client, 48
- Arc::SRMClient, 50
 - ~SRMClient, 52
 - abort, 52
 - cfg, 57
 - client, 57
 - copy, 52
 - getInstance, 52
 - getRequestTokens, 53
 - getSpaceTokens, 53
 - getURLs, 53
 - getVersion, 53
 - implementation, 57
 - info, 54
 - logger, 57
 - mkDir, 54
 - ns, 57
 - ping, 54
 - process, 55
 - putURLs, 55
 - release, 55
 - releaseGet, 55
 - releasePut, 56
 - remove, 56
 - request_timeout, 57
 - requestBringOnline, 56
 - requestBringOnlineStatus, 56
 - service_endpoint, 57
 - SRMClient, 52
 - Timeout, 57
 - user_timeout, 58
 - version, 58
- Arc::SRMClientRequest, 58
 - file_ids, 59
 - finished_success, 59
 - long_list, 59
 - request_id, 59
 - request_token, 59
 - space_token, 59
 - SRMClientRequest, 59
 - surl_failures, 59
 - surl_statuses, 59
 - surls, 60
 - waiting_time, 60
- Arc::SRMFileMetaData, 60
- Arc::SRMInvalidRequestException, 61
- ArcEvaluationCtx
 - ArcSec::ArcEvaluationCtx, 18
- ArcPolicy
 - ArcSec::ArcPolicy, 20
- ArcSec, 11
 - AndList, 14
 - Match, 14
- ArcSec::AllowPDP, 15
- ArcSec::ArcAlgFactory, 15
 - createAlg, 16
- ArcSec::ArcAttributeFactory, 16
 - createValue, 16
- ArcSec::ArcAttributeProxy, 16
- ArcSec::ArcAuthZ, 17
 - Handle, 17
 - MakePDPs, 17
- ArcSec::ArcEvaluationCtx, 18
 - ArcEvaluationCtx, 18
 - split, 18
- ArcSec::ArcEvaluator, 18
 - evaluate, 19
- ArcSec::ArcFnFactory, 19
 - createFn, 19
- ArcSec::ArcPDP, 19
- ArcSec::ArcPolicy, 20
 - ArcPolicy, 20
 - make_policy, 20
- ArcSec::ArcRequest, 20
- ArcSec::ArcRequestItem, 21
- ArcSec::ArcRequestTuple, 21
- ArcSec::ArcRule, 21
- ArcSec::AttributeDesignator, 21
- ArcSec::AttributeSelector, 22
- ArcSec::DelegationCollector, 23
- ArcSec::DelegationMultiSecAttr, 23

- ArcSec::DelegationPDP, 23
- ArcSec::DelegationSecAttr, 24
- ArcSec::DelegationSH, 24
- ArcSec::DenyPDP, 24
- ArcSec::GACLEvaluator, 24
 - evaluate, 24
- ArcSec::GACLPDP, 24
- ArcSec::GACLPolicy, 25
- ArcSec::GACLRequest, 25
- ArcSec::PDPServletInvoker, 40
- ArcSec::SAML2SSO_AssertionConsumerSH, 41
- ArcSec::SAMLTokenSH, 41
- ArcSec::SimpleListPDP, 41
- ArcSec::UsernameTokenSH, 62
- ArcSec::X509TokenSH, 62
- ArcSec::XACMLAlgFactory, 63
 - createAlg, 63
- ArcSec::XACMLApply, 63
- ArcSec::XACMLAttributeFactory, 63
 - createValue, 64
- ArcSec::XACMLAttributeProxy, 64
- ArcSec::XACMLCondition, 64
 - XACMLCondition, 65
- ArcSec::XACMLEvaluationCtx, 65
 - XACMLEvaluationCtx, 65
- ArcSec::XACMLEvaluator, 65
 - evaluate, 66
- ArcSec::XACMLFnFactory, 66
 - createFn, 66
- ArcSec::XACMLPDP, 67
- ArcSec::XACMLPolicy, 67
 - make_policy, 68
 - XACMLPolicy, 67
- ArcSec::XACMLRequest, 68
 - getEvalName, 68
 - getName, 68
- ArcSec::XACMLRule, 68
- ArcSec::XACMLTarget, 69
 - XACMLTarget, 69
- ArcSec::XACMLTargetMatch, 69
- ArcSec::XACMLTargetMatchGroup, 69
- ArcSec::XACMLTargetSection, 69
- Attribute
 - Arc::PayloadHTTP, 35
- Attributes
 - Arc::PayloadHTTP, 35
- attributes_
 - Arc::PayloadHTTP, 36
- BaseURL
 - SRMURL, 61
- Body
 - Arc::PayloadHTTP, 35
- body_own_
 - Arc::PayloadHTTP, 36
- cfg
 - Arc::SRMClient, 57
- chunked_
 - Arc::PayloadHTTP, 36
- client
 - Arc::SRMClient, 57
- code_
 - Arc::PayloadHTTP, 36
- ContactURL
 - SRMURL, 61
- copy
 - Arc::SRM1Client, 42
 - Arc::SRM22Client, 48
 - Arc::SRMClient, 52
- createAlg
 - ArcSec::ArcAlgFactory, 16
 - ArcSec::XACMLAlgFactory, 63
- createFn
 - ArcSec::ArcFnFactory, 19
 - ArcSec::XACMLFnFactory, 66
- createValue
 - ArcSec::ArcAttributeFactory, 16
 - ArcSec::XACMLAttributeFactory, 64
- Endpoint
 - SRMURL, 61
- evaluate
 - ArcSec::ArcEvaluator, 19
 - ArcSec::GACLEvaluator, 24
 - ArcSec::XACMLEvaluator, 66
- file_ids
 - Arc::SRMClientRequest, 59
- FileName
 - SRMURL, 61
- finished_success
 - Arc::SRMClientRequest, 59
- Flush
 - Arc::PayloadHTTP, 35
- FullURL
 - SRMURL, 62
- get_body
 - Arc::PayloadHTTP, 35
- GetCert
 - Arc::PayloadTLSStream, 40
- getEvalName
 - ArcSec::XACMLRequest, 68
- getInstance
 - Arc::SRMClient, 52
- getName
 - ArcSec::XACMLRequest, 68

- GetPeerCert
 - Arc::PayloadTLSStream, 40
- getRequestTokens
 - Arc::SRM1Client, 43
 - Arc::SRM22Client, 48
 - Arc::SRMClient, 53
- getSpaceTokens
 - Arc::SRM1Client, 43
 - Arc::SRM22Client, 48
 - Arc::SRMClient, 53
- getURLs
 - Arc::SRM1Client, 43
 - Arc::SRM22Client, 48
 - Arc::SRMClient, 53
- getVersion
 - Arc::SRMClient, 53
- Handle
 - ArcSec::ArcAuthZ, 17
- implementation
 - Arc::SRMClient, 57
- info
 - Arc::SRM1Client, 44
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 54
- keep_alive_
 - Arc::PayloadHTTP, 36
- LDAPQuery
 - Arc::LDAPQuery, 25
- length_
 - Arc::PayloadHTTP, 36
- logger
 - Arc::SRMClient, 57
- long_list
 - Arc::SRMClientRequest, 59
- make_policy
 - ArcSec::ArcPolicy, 20
 - ArcSec::XACMLPolicy, 68
- MakePDPs
 - ArcSec::ArcAuthZ, 17
- Match
 - ArcSec, 14
- MCC_TCP_Service
 - Arc::MCC_TCP_Service, 31
- method_
 - Arc::PayloadHTTP, 36
- mkDir
 - Arc::SRM1Client, 44
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 54
- ns
 - Arc::SRMClient, 57
- parse_header
 - Arc::PayloadHTTP, 35
- PayloadHTTP
 - Arc::PayloadHTTP, 34, 35
- PayloadTCPSocket
 - Arc::PayloadTCPSocket, 38
- PayloadTLMCC
 - Arc::PayloadTLMCC, 39
- PayloadTLSStream
 - Arc::PayloadTLSStream, 40
- ping
 - Arc::SRM1Client, 44
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 54
- PortDefined
 - SRMURL, 62
- process
 - Arc::SRMClient, 55
- putURLs
 - Arc::SRM1Client, 45
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 55
- Query
 - Arc::LDAPQuery, 26
- rbody_
 - Arc::PayloadHTTP, 36
- read
 - Arc::PayloadHTTP, 35
- readline
 - Arc::PayloadHTTP, 36
- reason_
 - Arc::PayloadHTTP, 36
- release
 - Arc::SRM1Client, 45
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 55
- releaseGet
 - Arc::SRM1Client, 45
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 55
- releasePut
 - Arc::SRM1Client, 46
 - Arc::SRM22Client, 49
 - Arc::SRMClient, 56
- remove
 - Arc::SRM1Client, 46
 - Arc::SRM22Client, 50
 - Arc::SRMClient, 56
- request_id

- Arc::SRMClientRequest, 59
- request_timeout
 - Arc::SRMClient, 57
- request_token
 - Arc::SRMClientRequest, 59
- requestBringOnline
 - Arc::SRM1Client, 46
 - Arc::SRM22Client, 50
 - Arc::SRMClient, 56
- requestBringOnlineStatus
 - Arc::SRM1Client, 47
 - Arc::SRM22Client, 50
 - Arc::SRMClient, 56
- Result
 - Arc::LDAPQuery, 26
- sbody_
 - Arc::PayloadHTTP, 36
- service_endpoint
 - Arc::SRMClient, 57
- SetSRMVersion
 - SRMURL, 62
- ShortURL
 - SRMURL, 62
- space_token
 - Arc::SRMClientRequest, 59
- split
 - ArcSec::ArcEvaluationCtx, 18
- SRM22Client
 - Arc::SRM22Client, 48
- SRMClient
 - Arc::SRMClient, 52
- SRMClientRequest
 - Arc::SRMClientRequest, 59
- SRMFileInfo, 60
- SRMInfo, 60
- SRMURL, 61
 - BaseURL, 61
 - ContactURL, 61
 - Endpoint, 61
 - FileName, 61
 - FullURL, 62
 - PortDefined, 62
 - SetSRMVersion, 62
 - ShortURL, 62
 - SRMURL, 61
- ssl_
 - Arc::PayloadTLSStream, 40
- STACK_OF
 - Arc::PayloadTLSStream, 40
- stream_
 - Arc::PayloadHTTP, 37
- stream_own_
 - Arc::PayloadHTTP, 37
- surl_failures
 - Arc::SRMClientRequest, 59
- surl_statuses
 - Arc::SRMClientRequest, 59
- surls
 - Arc::SRMClientRequest, 60
- Timeout
 - Arc::SRMClient, 57
- uri_
 - Arc::PayloadHTTP, 37
- user_timeout
 - Arc::SRMClient, 58
- version
 - Arc::SRMClient, 58
- version_major_
 - Arc::PayloadHTTP, 37
- version_minor_
 - Arc::PayloadHTTP, 37
- waiting_time
 - Arc::SRMClientRequest, 60
- XACMLCondition
 - ArcSec::XACMLCondition, 65
- XACMLEvaluationCtx
 - ArcSec::XACMLEvaluationCtx, 65
- XACMLPolicy
 - ArcSec::XACMLPolicy, 67
- XACMLTarget
 - ArcSec::XACMLTarget, 69