

Hosting Environment (Daemon) Chain Components

Generated by Doxygen 1.7.4

Sun Sep 25 2011 05:46:21

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	13
4.1.2.1	AndList	13
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	18
5.5.2.1	Handle	18
5.5.2.2	MakePDPs	18
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	19
5.7.1	Detailed Description	19
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	20
5.8.2.1	createFn	20
5.9	ArcSec::ArcPDP Class Reference	20
5.9.1	Detailed Description	20
5.10	ArcSec::ArcPolicy Class Reference	20
5.10.1	Detailed Description	20
5.10.2	Constructor & Destructor Documentation	21
5.10.2.1	ArcPolicy	21
5.10.2.2	ArcPolicy	21
5.10.2.3	ArcPolicy	21
5.10.3	Member Function Documentation	21
5.10.3.1	make_policy	21
5.11	ArcSec::ArcRequest Class Reference	21
5.12	ArcSec::ArcRequestItem Class Reference	21
5.12.1	Detailed Description	21

5.13 ArcSec::ArcRequestTuple Class Reference	22
5.13.1 Detailed Description	22
5.14 ArcSec::ArcRule Class Reference	22
5.14.1 Detailed Description	22
5.15 ArcSec::AttributeDesignator Class Reference	22
5.16 ArcSec::AttributeSelector Class Reference	22
5.17 ArcSHCLegacy::AuthUser Class Reference	22
5.18 ArcSHCLegacy::AuthVO Class Reference	23
5.19 ArcSHCLegacy::ConfigParser Class Reference	23
5.20 Arc::ConfigTlsmcc Class Reference	23
5.21 Arc::DataPointARC Class Reference	23
5.21.1 Detailed Description	23
5.22 Arc::DataPointFile Class Reference	24
5.22.1 Detailed Description	24
5.23 Arc::DataPointGridFTP Class Reference	24
5.23.1 Detailed Description	24
5.24 Arc::DataPointHTTP Class Reference	24
5.24.1 Detailed Description	24
5.25 Arc::DataPointLDAP Class Reference	25
5.25.1 Detailed Description	25
5.26 Arc::DataPointLFC Class Reference	25
5.26.1 Detailed Description	25
5.27 Arc::DataPointRLS Class Reference	25
5.27.1 Detailed Description	26
5.28 Arc::DataPointSRM Class Reference	26
5.28.1 Detailed Description	26
5.29 Arc::DataPointXrootd Class Reference	26
5.29.1 Detailed Description	26
5.30 ArcSec::DelegationCollector Class Reference	27
5.31 ArcSec::DelegationMultiSecAttr Class Reference	27
5.32 ArcSec::DelegationPDP Class Reference	27
5.32.1 Detailed Description	27
5.33 ArcSec::DelegationSecAttr Class Reference	27
5.34 ArcSec::DelegationSH Class Reference	27

5.35 ArcSec::DenyPDP Class Reference	27
5.35.1 Detailed Description	28
5.36 ArcSec::GACLEvaluator Class Reference	28
5.36.1 Member Function Documentation	28
5.36.1.1 evaluate	28
5.37 ArcSec::GACLPDP Class Reference	28
5.38 ArcSec::GACLPolicy Class Reference	28
5.39 ArcSec::GACLRequest Class Reference	28
5.40 Arc::LDAPQuery Class Reference	29
5.40.1 Detailed Description	29
5.40.2 Constructor & Destructor Documentation	29
5.40.2.1 LDAPQuery	29
5.40.2.2 ~LDAPQuery	29
5.40.3 Member Function Documentation	29
5.40.3.1 Query	29
5.40.3.2 Result	29
5.41 ArcSHCLegacy::LegacyMap Class Reference	30
5.42 ArcSHCLegacy::LegacyPDP Class Reference	30
5.43 ArcSHCLegacy::LegacySecAttr Class Reference	30
5.44 ArcSHCLegacy::LegacySecHandler Class Reference	30
5.45 Arc::Lister Class Reference	30
5.46 Arc::MCC_GSI_Client Class Reference	31
5.47 Arc::MCC_GSI_Service Class Reference	31
5.48 Arc::MCC_HTTP Class Reference	31
5.48.1 Detailed Description	31
5.49 Arc::MCC_HTTP_Client Class Reference	31
5.49.1 Detailed Description	32
5.50 Arc::MCC_HTTP_Service Class Reference	32
5.50.1 Detailed Description	32
5.51 Arc::MCC_MsgValidator Class Reference	33
5.52 Arc::MCC_MsgValidator_Service Class Reference	33
5.53 Arc::MCC_SOAP Class Reference	33
5.53.1 Detailed Description	34
5.54 Arc::MCC_SOAP_Client Class Reference	34

5.55 Arc::MCC_SOAP_Service Class Reference	34
5.55.1 Detailed Description	35
5.56 Arc::MCC_TCP Class Reference	35
5.56.1 Detailed Description	35
5.57 Arc::MCC_TCP_Client Class Reference	35
5.57.1 Detailed Description	36
5.58 Arc::MCC_TCP_Service Class Reference	36
5.58.1 Detailed Description	36
5.58.2 Constructor & Destructor Documentation	37
5.58.2.1 MCC_TCP_Service	37
5.59 Arc::MCC_TLS Class Reference	37
5.59.1 Detailed Description	37
5.60 Arc::MCC_TLS_Client Class Reference	37
5.60.1 Detailed Description	38
5.61 Arc::MCC_TLS_Service Class Reference	38
5.61.1 Detailed Description	38
5.62 Arc::PayloadGSISStream Class Reference	38
5.63 Arc::PayloadHTTP Class Reference	39
5.63.1 Detailed Description	40
5.63.2 Constructor & Destructor Documentation	40
5.63.2.1 PayloadHTTP	40
5.63.2.2 PayloadHTTP	40
5.63.2.3 PayloadHTTP	40
5.63.2.4 PayloadHTTP	40
5.63.2.5 PayloadHTTP	40
5.63.3 Member Function Documentation	40
5.63.3.1 Attribute	40
5.63.3.2 Attribute	41
5.63.3.3 Attributes	41
5.63.3.4 Body	41
5.63.3.5 Flush	41
5.63.3.6 get_body	41
5.63.3.7 parse_header	41
5.63.3.8 read	41

5.63.3.9	readline	41
5.63.4	Field Documentation	41
5.63.4.1	attributes_	42
5.63.4.2	body_own_	42
5.63.4.3	chunked_	42
5.63.4.4	code_	42
5.63.4.5	keep_alive_	42
5.63.4.6	length_	42
5.63.4.7	method_	42
5.63.4.8	rbody_	42
5.63.4.9	reason_	42
5.63.4.10	sbody_	42
5.63.4.11	stream_	43
5.63.4.12	stream_own_	43
5.63.4.13	uri_	43
5.63.4.14	version_major_	43
5.63.4.15	version_minor_	43
5.64	Arc::PayloadTCPSocket Class Reference	43
5.64.1	Detailed Description	43
5.64.2	Constructor & Destructor Documentation	44
5.64.2.1	PayloadTCPSocket	44
5.64.2.2	PayloadTCPSocket	44
5.64.2.3	PayloadTCPSocket	44
5.64.2.4	PayloadTCPSocket	44
5.64.2.5	PayloadTCPSocket	44
5.65	Arc::PayloadTLMCC Class Reference	44
5.65.1	Constructor & Destructor Documentation	45
5.65.1.1	PayloadTLMCC	45
5.65.1.2	PayloadTLMCC	45
5.65.1.3	PayloadTLMCC	45
5.66	Arc::PayloadTLSStream Class Reference	45
5.66.1	Detailed Description	46
5.66.2	Constructor & Destructor Documentation	46
5.66.2.1	PayloadTLSStream	46

5.66.2.2	~PayloadTLSStream	46
5.66.3	Member Function Documentation	46
5.66.3.1	GetCert	46
5.66.3.2	GetPeerCert	46
5.66.3.3	STACK_OF	46
5.66.4	Field Documentation	47
5.66.4.1	ssl_	47
5.67	ArcSec::PDPServletInvoker Class Reference	47
5.67.1	Detailed Description	47
5.68	ArcSec::SAML2SSO_AssertionConsumerSH Class Reference	47
5.68.1	Detailed Description	47
5.69	ArcSec::SAMLTokenSH Class Reference	47
5.69.1	Detailed Description	48
5.70	ArcSec::SimpleListPDP Class Reference	48
5.70.1	Detailed Description	48
5.71	ArcSHCLegacy::SimpleMap Class Reference	48
5.72	Arc::SRM1Client Class Reference	48
5.72.1	Member Function Documentation	49
5.72.1.1	abort	49
5.72.1.2	checkPermissions	50
5.72.1.3	copy	50
5.72.1.4	getRequestTokens	50
5.72.1.5	getSpaceTokens	51
5.72.1.6	getTURLs	51
5.72.1.7	getTURLsStatus	51
5.72.1.8	info	52
5.72.1.9	mkDir	52
5.72.1.10	ping	52
5.72.1.11	putTURLs	53
5.72.1.12	putTURLsStatus	53
5.72.1.13	release	54
5.72.1.14	releaseGet	54
5.72.1.15	releasePut	54
5.72.1.16	remove	54

5.72.1.17 requestBringOnline	55
5.72.1.18 requestBringOnlineStatus	55
5.73 Arc::SRM22Client Class Reference	56
5.73.1 Constructor & Destructor Documentation	56
5.73.1.1 SRM22Client	56
5.73.1.2 ~SRM22Client	57
5.73.2 Member Function Documentation	57
5.73.2.1 abort	57
5.73.2.2 checkPermissions	57
5.73.2.3 copy	57
5.73.2.4 getRequestTokens	57
5.73.2.5 getSpaceTokens	57
5.73.2.6 getURLs	57
5.73.2.7 getURLsStatus	58
5.73.2.8 info	58
5.73.2.9 mkDir	58
5.73.2.10 ping	58
5.73.2.11 putURLs	58
5.73.2.12 putURLsStatus	58
5.73.2.13 release	59
5.73.2.14 releaseGet	59
5.73.2.15 releasePut	59
5.73.2.16 remove	59
5.73.2.17 requestBringOnline	59
5.73.2.18 requestBringOnlineStatus	59
5.74 Arc::SRMClient Class Reference	60
5.74.1 Detailed Description	61
5.74.2 Constructor & Destructor Documentation	61
5.74.2.1 SRMClient	61
5.74.2.2 ~SRMClient	61
5.74.3 Member Function Documentation	61
5.74.3.1 abort	61
5.74.3.2 checkPermissions	62
5.74.3.3 copy	62

5.74.3.4	getInstance	62
5.74.3.5	getRequestTokens	63
5.74.3.6	getSpaceTokens	63
5.74.3.7	getTURLs	64
5.74.3.8	getTURLsStatus	64
5.74.3.9	getVersion	64
5.74.3.10	info	64
5.74.3.11	mkDir	65
5.74.3.12	ping	65
5.74.3.13	process	66
5.74.3.14	putTURLs	66
5.74.3.15	putTURLsStatus	66
5.74.3.16	release	66
5.74.3.17	releaseGet	67
5.74.3.18	releasePut	67
5.74.3.19	remove	67
5.74.3.20	requestBringOnline	68
5.74.3.21	requestBringOnlineStatus	68
5.74.4	Field Documentation	68
5.74.4.1	cfg	68
5.74.4.2	client	69
5.74.4.3	implementation	69
5.74.4.4	logger	69
5.74.4.5	ns	69
5.74.4.6	service_endpoint	69
5.74.4.7	user_timeout	69
5.74.4.8	version	69
5.75	Arc::SRMClientRequest Class Reference	69
5.75.1	Detailed Description	70
5.75.2	Constructor & Destructor Documentation	70
5.75.2.1	SRMClientRequest	70
5.75.2.2	SRMClientRequest	70
5.75.3	Member Function Documentation	70
5.75.3.1	file_ids	70

5.75.3.2	finished_success	70
5.75.3.3	long_list	71
5.75.3.4	request_id	71
5.75.3.5	request_timeout	71
5.75.3.6	request_token	71
5.75.3.7	space_token	71
5.75.3.8	surl_failures	71
5.75.3.9	surl_statuses	71
5.75.3.10	surls	71
5.75.3.11	total_size	71
5.75.3.12	transport_protocols	71
5.75.3.13	waiting_time	72
5.76	SRMFileInfo Class Reference	72
5.76.1	Detailed Description	72
5.77	Arc::SRMFileMetaData Struct Reference	72
5.77.1	Detailed Description	72
5.78	SRMInfo Class Reference	72
5.78.1	Detailed Description	72
5.79	Arc::SRMInvalidRequestException Class Reference	73
5.80	SRMURL Class Reference	73
5.80.1	Constructor & Destructor Documentation	73
5.80.1.1	SRMURL	73
5.80.2	Member Function Documentation	73
5.80.2.1	BaseURL	73
5.80.2.2	ContactURL	73
5.80.2.3	Endpoint	73
5.80.2.4	FileName	73
5.80.2.5	FullURL	74
5.80.2.6	PortDefined	74
5.80.2.7	SetSRMVersion	74
5.80.2.8	ShortURL	74
5.81	ArcSHCLegacy::UnixMap Class Reference	74
5.82	ArcSec::UsernameTokenSH Class Reference	74
5.82.1	Detailed Description	74

5.83 ArcSHCLegacy::voms Struct Reference	75
5.83.1 Detailed Description	75
5.83.2 Field Documentation	75
5.83.2.1 attrs	75
5.83.2.2 server	75
5.83.2.3 voname	75
5.84 ArcSHCLegacy::voms_attrs Struct Reference	75
5.84.1 Detailed Description	76
5.84.2 Field Documentation	76
5.84.2.1 cap	76
5.84.2.2 group	76
5.84.2.3 role	76
5.85 ArcSec::X509TokenSH Class Reference	76
5.85.1 Detailed Description	76
5.86 ArcSec::XACMLAlgFactory Class Reference	76
5.86.1 Detailed Description	77
5.86.2 Member Function Documentation	77
5.86.2.1 createAlg	77
5.87 ArcSec::XACMLApply Class Reference	77
5.88 ArcSec::XACMLAttributeFactory Class Reference	77
5.88.1 Detailed Description	77
5.88.2 Member Function Documentation	78
5.88.2.1 createValue	78
5.89 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Refer- ence	78
5.89.1 Detailed Description	78
5.90 ArcSec::XACMLCondition Class Reference	78
5.90.1 Detailed Description	79
5.90.2 Constructor & Destructor Documentation	79
5.90.2.1 XACMLCondition	79
5.91 ArcSec::XACMLEvaluationCtx Class Reference	79
5.91.1 Detailed Description	79
5.91.2 Constructor & Destructor Documentation	79
5.91.2.1 XACMLEvaluationCtx	79

5.92 ArcSec::XACMLEvaluator Class Reference	79
5.92.1 Detailed Description	80
5.92.2 Member Function Documentation	80
5.92.2.1 evaluate	80
5.93 ArcSec::XACMLFnFactory Class Reference	80
5.93.1 Detailed Description	80
5.93.2 Member Function Documentation	80
5.93.2.1 createFn	80
5.94 ArcSec::XACMLPDP Class Reference	81
5.94.1 Detailed Description	81
5.95 ArcSec::XACMLPolicy Class Reference	81
5.95.1 Detailed Description	81
5.95.2 Constructor & Destructor Documentation	81
5.95.2.1 XACMLPolicy	81
5.95.2.2 XACMLPolicy	81
5.95.2.3 XACMLPolicy	82
5.95.3 Member Function Documentation	82
5.95.3.1 make_policy	82
5.96 ArcSec::XACMLRequest Class Reference	82
5.96.1 Member Function Documentation	82
5.96.1.1 getEvalName	82
5.96.1.2 getName	82
5.97 ArcSec::XACMLRule Class Reference	82
5.97.1 Detailed Description	83
5.98 ArcSec::XACMLTarget Class Reference	83
5.98.1 Detailed Description	83
5.98.2 Constructor & Destructor Documentation	83
5.98.2.1 XACMLTarget	83
5.99 ArcSec::XACMLTargetMatch Class Reference	83
5.100 ArcSec::XACMLTargetMatchGroup Class Reference	83
5.101 ArcSec::XACMLTargetSection Class Reference	84

Chapter 1

Namespace Index

1.1 Namespace List

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

[ArcSec](#) ([ArcRequest](#), 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	19
ArcSec::ArcFnFactory	19
ArcSec::ArcPDP	20
ArcSec::ArcPolicy	20
ArcSec::ArcRequest	21
ArcSec::ArcRequestItem	21
ArcSec::ArcRequestTuple	22
ArcSec::ArcRule	22
ArcSec::AttributeDesignator	22
ArcSec::AttributeSelector	22
ArcSHCLegacy::AuthUser	22
ArcSHCLegacy::AuthVO	23
ArcSHCLegacy::ConfigParser	23
Arc::ConfigTLMCC	23
Arc::DataPointARC	23
Arc::DataPointFile	24
Arc::DataPointGridFTP	24
Arc::DataPointHTTP	24
Arc::DataPointLDAP	25
Arc::DataPointLFC	25
Arc::DataPointRLS	25
Arc::DataPointSRM	26
Arc::DataPointXrootd	26

ArcSec::DelegationCollector	27
ArcSec::DelegationMultiSecAttr	27
ArcSec::DelegationPDP	27
ArcSec::DelegationSecAttr	27
ArcSec::DelegationSH	27
ArcSec::DenyPDP	27
ArcSec::GACLEvaluator	28
ArcSec::GACLPDP	28
ArcSec::GACLPolicy	28
ArcSec::GACLRequest	28
Arc::LDAPQuery	29
ArcSHCLegacy::LegacyMap	30
ArcSHCLegacy::LegacyPDP	30
ArcSHCLegacy::LegacySecAttr	30
ArcSHCLegacy::LegacySecHandler	30
Arc::Lister	30
Arc::MCC_GSI_Client	31
Arc::MCC_GSI_Service	31
Arc::MCC_HTTP	31
Arc::MCC_HTTP_Client	31
Arc::MCC_HTTP_Service	32
Arc::MCC_MsgValidator	33
Arc::MCC_MsgValidator_Service	33
Arc::MCC_SOAP	33
Arc::MCC_SOAP_Client	34
Arc::MCC_SOAP_Service	34
Arc::MCC_TCP	35
Arc::MCC_TCP_Client	35
Arc::MCC_TCP_Service	36
Arc::MCC_TLS	37
Arc::MCC_TLS_Client	37
Arc::MCC_TLS_Service	38
Arc::PayloadGSISStream	38
Arc::PayloadHTTP	39
Arc::PayloadTCPsocket	43
Arc::PayloadTLSStream	45
Arc::PayloadTLSMCC	44
ArcSec::PDPServletInvoker	47
ArcSec::SAML2SSO_AssertionConsumerSH	47
ArcSec::SAMLTokenSH	47
ArcSec::SimpleListPDP	48
ArcSHCLegacy::SimpleMap	48
Arc::SRMClient	60
Arc::SRM1Client	48
Arc::SRM22Client	56
Arc::SRMClientRequest	69
SRMFileInfo	72
Arc::SRMFileMetaData	72

SRMInfo	72
Arc::SRMInvalidRequestException	73
SRMURL	73
ArcSHCLegacy::UnixMap	74
ArcSec::UsernameTokenSH	74
ArcSHCLegacy::voms	75
ArcSHCLegacy::voms_attrs	75
ArcSec::X509TokenSH	76
ArcSec::XACMLAlgFactory	76
ArcSec::XACMLApply	77
ArcSec::XACMLAttributeFactory	77
ArcSec::XACMLAttributeProxy< TheAttribute >	78
ArcSec::XACMLCondition	78
ArcSec::XACMLEvaluationCtx	79
ArcSec::XACMLEvaluator	79
ArcSec::XACMLFnFactory	80
ArcSec::XACMLPDP	81
ArcSec::XACMLPolicy	81
ArcSec::XACMLRequest	82
ArcSec::XACMLRule	82
ArcSec::XACMLTarget	83
ArcSec::XACMLTargetMatch	83
ArcSec::XACMLTargetMatchGroup	83
ArcSec::XACMLTargetSection	84

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)	19
ArcSec::ArcFnFactory (Function factory class for Arc specified attributes) . . .	19
ArcSec::ArcPDP (ArcPDP - PDP which can handle the Arc specific request and policy schema)	20
ArcSec::ArcPolicy (ArcPolicy class to parse and operate Arc specific <Policy> node)	20
ArcSec::ArcRequest	21
ArcSec::ArcRequestItem (Container, <Subjects, Actions, Objects, Contexts> tuple)	21
ArcSec::ArcRequestTuple (RequestTuple, container which includes the) . . .	22
ArcSec::ArcRule (ArcRule class to parse Arc specific <Rule> node)	22
ArcSec::AttributeDesignator	22
ArcSec::AttributeSelector	22
ArcSHCLegacy::AuthUser	22
ArcSHCLegacy::AuthVO	23
ArcSHCLegacy::ConfigParser	23
Arc::ConfigTLMCC	23
Arc::DataPointARC	23
Arc::DataPointFile	24
Arc::DataPointGridFTP	24
Arc::DataPointHTTP	24

Arc::DataPointLDAP	25
Arc::DataPointLFC	25
Arc::DataPointRLS	25
Arc::DataPointSRM	26
Arc::DataPointXrootd	26
ArcSec::DelegationCollector	27
ArcSec::DelegationMultiSecAttr	27
ArcSec::DelegationPDP	27
ArcSec::DelegationSecAttr	27
ArcSec::DelegationSH	27
ArcSec::DenyPDP (This PDP always returns false (deny))	27
ArcSec::GACLEvaluator	28
ArcSec::GACLPDP	28
ArcSec::GACLPolicy	28
ArcSec::GACLRequest	28
Arc::LDAPQuery	29
ArcSHCLegacy::LegacyMap	30
ArcSHCLegacy::LegacyPDP	30
ArcSHCLegacy::LegacySecAttr	30
ArcSHCLegacy::LegacySecHandler	30
Arc::Lister	30
Arc::MCC_GSI_Client	31
Arc::MCC_GSI_Service	31
Arc::MCC_HTTP (A base class for HTTP client and service MCCs)	31
Arc::MCC_HTTP_Client	31
Arc::MCC_HTTP_Service	32
Arc::MCC_MsgValidator	33
Arc::MCC_MsgValidator_Service	33
Arc::MCC_SOAP (A base class for SOAP client and service MCCs)	33
Arc::MCC_SOAP_Client	34
Arc::MCC_SOAP_Service	34
Arc::MCC_TCP (A base class for TCP client and service MCCs)	35
Arc::MCC_TCP_Client	35
Arc::MCC_TCP_Service	36
Arc::MCC_TLS (A base class for TLS client and service MCCs)	37
Arc::MCC_TLS_Client	37
Arc::MCC_TLS_Service	38
Arc::PayloadGSISStream	38
Arc::PayloadHTTP	39
Arc::PayloadTCPSocket	43
Arc::PayloadTLMCC	44
Arc::PayloadTLSSStream	45
ArcSec::PDPServiceInvoker (PDPServiceInvoker - client which will invoke pdpser-vice)	47
ArcSec::SAML2SSO_AssertionConsumerSH (Implement the functionality of the Service Provider in SAML2 SSO profile)	47
ArcSec::SAMLTokenSH (Adds WS-Security SAML Token into SOAP Header)	47
ArcSec::SimpleListPDP (Tests X509 subject against list of subjects in file)	48
ArcSHCLegacy::SimpleMap	48
Arc::SRM1Client	48

Arc::SRM22Client	56
Arc::SRMClient	60
Arc::SRMClientRequest	69
SRMFileInfo	72
Arc::SRMFileMetaData	72
SRMInfo	72
Arc::SRMInvalidRequestException	73
SRMURL	73
ArcSHCLegacy::UnixMap	74
ArcSec::UsernameTokenSH (Adds WS-Security Username Token into SOAP Header)	74
ArcSHCLegacy::voms	75
ArcSHCLegacy::voms_attrs	75
ArcSec::X509TokenSH (Adds WS-Security X509 Token into SOAP Header)	76
ArcSec::XACMLAlgFactory (Algorithm factory class for XACML)	76
ArcSec::XACMLApply	77
ArcSec::XACMLAttributeFactory (Attribute factory class for XACML specified attributes)	77
ArcSec::XACMLAttributeProxy< TheAttribute > (XACML specific AttributeProxy class)	78
ArcSec::XACMLCondition (XACMLCondition class to parse and operate XACML specific <Condition> node)	78
ArcSec::XACMLEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	79
ArcSec::XACMLEvaluator (Execute the policy evaluation, based on the request and policy)	79
ArcSec::XACMLFnFactory (Function factory class for XACML specified attributes)	80
ArcSec::XACMLPDP (XACMLPDP - PDP which can handle the XACML specific request and policy schema)	81
ArcSec::XACMLPolicy (XACMLPolicy class to parse and operate XACML specific <Policy> node)	81
ArcSec::XACMLRequest	82
ArcSec::XACMLRule (XACMLRule class to parse XACML specific <Rule> node)	82
ArcSec::XACMLTarget (XACMLTarget class to parse and operate XACML specific <Target> node)	83
ArcSec::XACMLTargetMatch	83
ArcSec::XACMLTargetMatchGroup	83
ArcSec::XACMLTargetSection	84

Chapter 4

Namespace Documentation

4.1 ArcSec Namespace Reference

[ArcRequest](#), Parsing the specified Arc request format.

Data Structures

- class [DelegationCollector](#)
- class [DelegationSecAttr](#)
- class [DelegationMultiSecAttr](#)
- class [AllowPDP](#)
This PDP always return true (allow)
- class [ArcAuthZ](#)
Tests message against list of PDPs.
- class [ArcAlgFactory](#)
Algorithm factory class for Arc.
- class [ArcAttributeFactory](#)
Attribute factory class for Arc specified attributes.
- class [ArcAttributeProxy](#)
Arc specific AttributeProxy class.
- class [ArcRequestTuple](#)
RequestTuple, container which includes the.
- class [ArcEvaluationCtx](#)
EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.
- class [ArcEvaluator](#)
Execute the policy evaluation, based on the request and policy.
- class [ArcFnFactory](#)
Function factory class for Arc specified attributes.
- class [ArcPDP](#)

ArcPDP - PDP which can handle the Arc specific request and policy schema.

- class [ArcPolicy](#)

ArcPolicy class to parse and operate Arc specific <Policy> node.

- class [ArcRequest](#)
- class [ArcRequestItem](#)

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

- class [ArcRule](#)

ArcRule 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 - 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 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 - PDP which can handle the XACML specific request and policy schema.
- class [XACMLPolicy](#)
XACMLPolicy class to parse and operate XACML specific <Policy> node.
- class [XACMLRequest](#)
- class [XACMLRule](#)
XACMLRule class to parse XACML specific <Rule> node.
- class [XACMLTargetMatch](#)
- class [XACMLTargetMatchGroup](#)
- class [XACMLTargetSection](#)
- class [XACMLTarget](#)
XACMLTarget class to parse and operate XACML specific <Target> node.

Typedefs

- typedef std::pair< AttributeValue *, Function * > [Match](#)
- typedef std::list< [Match](#) > [AndList](#)
- typedef std::list< [AndList](#) > [OrList](#)

4.1.1 Detailed Description

[ArcRequest](#), Parsing the specified Arc request format. [XACMLRequest](#), 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:ex</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/user</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 type="string">/vo.knowarc/usergroupA</Subject> <Subject> <SubFraction type="string">/O=Grid/OU=KnowARC/CN=XYZ</SubFraction> <SubFraction type="string">urn:mace:shibboleth:1.0:/O=Grid/OU=KnowARC/CN=XYZ</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](#) 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\(\)](#) method runs provided Message instance against all PDPs specified in configuration. If any of PDPs returns positive result [Handle\(\)](#) 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](#) 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](#) - PDP which can handle the Arc specific request and policy schema.

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

5.9.1 Detailed Description

[ArcPDP](#) - 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](#) 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](#) 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](#) which can parse Arc request formate

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](#) class to parse Arc specific <Rule> node.

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

5.14.1 Detailed Description

[ArcRule](#) 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 ArcSHCLegacy::AuthUser Class Reference

Data Structures

- class **group_t**
- struct **source_t**

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

- auth.h

5.18 ArcSHCLegacy::AuthVO Class Reference

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

- auth.h

5.19 ArcSHCLegacy::ConfigParser Class Reference

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

- ConfigParser.h

5.20 Arc::ConfigTLMCC Class Reference

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

- ConfigTLMCC.h

5.21 Arc::DataPointARC Class Reference

```
#include <DataPointARC.h>
```

5.21.1 Detailed Description

Provides an interface to the Chelonia storage system developed by ARC.

This class is a loadable module and cannot be used directly. The DataHandle class loads modules at runtime and should be used instead of this.

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

- DataPointARC.h

5.22 Arc::DataPointFile Class Reference

```
#include <DataPointFile.h>
```

5.22.1 Detailed Description

This class allows access to the regular local filesystem through the same interface as is used for remote storage on the grid.

This class is a loadable module and cannot be used directly. The DataHandle class loads modules at runtime and should be used instead of this.

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

- DataPointFile.h

5.23 Arc::DataPointGridFTP Class Reference

```
#include <DataPointGridFTP.h>
```

Data Structures

- class **CBAArg**

5.23.1 Detailed Description

GridFTP is essentially the FTP protocol with GSI security. This class uses libraries from the Globus Toolkit. It can also be used for regular FTP.

This class is a loadable module and cannot be used directly. The DataHandle class loads modules at runtime and should be used instead of this.

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

- DataPointGridFTP.h

5.24 Arc::DataPointHTTP Class Reference

```
#include <DataPointHTTP.h>
```

5.24.1 Detailed Description

This class allows access through HTTP to remote resources. HTTP over SSL (HTTPS) and HTTP over GSI (HTTPG) are also supported.

This class is a loadable module and cannot be used directly. The DataHandle class loads modules at runtime and should be used instead of this.

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

- DataPointHTTP.h

5.25 Arc::DataPointLDAP Class Reference

```
#include <DataPointLDAP.h>
```

5.25.1 Detailed Description

LDAP is used in grids mainly to store information about grid services or resources rather than to store data itself. This class allows access to LDAP data through the same interface as other grid resources.

This class is a loadable module and cannot be used directly. The DataHandle class loads modules at runtime and should be used instead of this.

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

- DataPointLDAP.h

5.26 Arc::DataPointLFC Class Reference

```
#include <DataPointLFC.h>
```

5.26.1 Detailed Description

The LCG File Catalog (LFC) is a replica catalog developed by CERN. It consists of a hierarchical namespace of grid files and each filename can be associated with one or more physical locations.

This class is a loadable module and cannot be used directly. The DataHandle class loads modules at runtime and should be used instead of this.

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

- DataPointLFC.h

5.27 Arc::DataPointRLS Class Reference

```
#include <DataPointRLS.h>
```

5.27.1 Detailed Description

The Replica Location Service (RLS) is a replica catalog developed by Globus. It maps filenames in a flat namespace to one or more physical locations, and can also store meta-information on each file. This class uses the Globus Toolkit libraries for accessing RLS.

This class is a loadable module and cannot be used directly. The `DataHandle` class loads modules at runtime and should be used instead of this.

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

- `DataPointRLS.h`

5.28 Arc::DataPointSRM Class Reference

```
#include <DataPointSRM.h>
```

5.28.1 Detailed Description

The Storage Resource Manager (SRM) protocol allows access to data distributed across physical storage through a unified namespace and management interface. `PrepareReading()` or `PrepareWriting()` must be used before reading or writing a physical file.

This class is a loadable module and cannot be used directly. The `DataHandle` class loads modules at runtime and should be used instead of this.

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

- `DataPointSRM.h`

5.29 Arc::DataPointXrootd Class Reference

```
#include <DataPointXrootd.h>
```

5.29.1 Detailed Description

xrootd is a protocol for data access across large scale storage clusters. More information can be found at <http://xrootd.slac.stanford.edu/>

This class is a loadable module and cannot be used directly. The `DataHandle` class loads modules at runtime and should be used instead of this.

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

- `DataPointXrootd.h`

5.30 ArcSec::DelegationCollector Class Reference

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

- DelegationCollector.h

5.31 ArcSec::DelegationMultiSecAttr Class Reference

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

- DelegationSecAttr.h

5.32 ArcSec::DelegationPDP Class Reference

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

5.32.1 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

5.33 ArcSec::DelegationSecAttr Class Reference

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

- DelegationSecAttr.h

5.34 ArcSec::DelegationSH Class Reference

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

- DelegationSH.h

5.35 ArcSec::DenyPDP Class Reference

This PDP always returns false (deny)

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

5.35.1 Detailed Description

This PDP always returns false (deny)

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

- DenyPDP.h

5.36 ArcSec::GACLEvaluator Class Reference

Public Member Functions

- virtual Response * [evaluate](#) (Request *request)

5.36.1 Member Function Documentation

5.36.1.1 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

5.37 ArcSec::GACLPDP Class Reference

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

- GACLPDP.h

5.38 ArcSec::GACLPolicy Class Reference

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

- GACLPolicy.h

5.39 ArcSec::GACLRequest Class Reference

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

- GACLRequest.h

5.40 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)

5.40.1 Detailed Description

[LDAPQuery](#) class; querying of LDAP servers.

5.40.2 Constructor & Destructor Documentation

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

Constructs a new [LDAPQuery](#) object and sets connection options. The connection is first established when calling [Query](#).

5.40.2.2 `Arc::LDAPQuery::~~LDAPQuery ()`

Destructor. Will disconnect from the ldapserver if still connected.

5.40.3 Member Function Documentation

5.40.3.1 `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.

5.40.3.2 `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

5.41 ArcSHCLegacy::LegacyMap Class Reference

Data Structures

- class **cfgfile**

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

- LegacyMap.h

5.42 ArcSHCLegacy::LegacyPDP Class Reference

Data Structures

- class **cfgfile**

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

- LegacyPDP.h

5.43 ArcSHCLegacy::LegacySecAttr Class Reference

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

- LegacySecAttr.h

5.44 ArcSHCLegacy::LegacySecHandler Class Reference

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

- LegacySecHandler.h

5.45 Arc::Lister Class Reference

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

- Lister.h

5.46 Arc::MCC_GSI_Client Class Reference

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

- MCCGSI.h

5.47 Arc::MCC_GSI_Service Class Reference

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

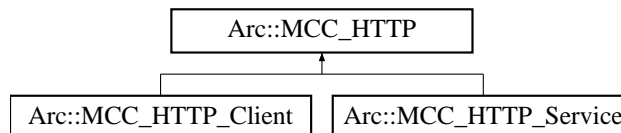
- MCCGSI.h

5.48 Arc::MCC_HTTP Class Reference

A base class for HTTP client and service MCCs.

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

Inheritance diagram for Arc::MCC_HTTP:



5.48.1 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.

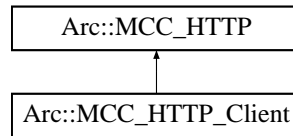
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.49 Arc::MCC_HTTP_Client Class Reference

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

Inheritance diagram for Arc::MCC_HTTP_Client:



5.49.1 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.

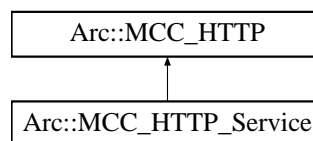
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.50 Arc::MCC_HTTP_Service Class Reference

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

Inheritance diagram for Arc::MCC_HTTP_Service:



5.50.1 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](#). 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

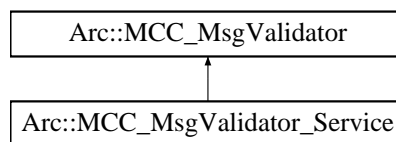
byte range HTTP:RANGEEND - end of requested byte range (inclusive) HTTP:name - all 'name' attributes of HTTP header. Attributes of response message of HTTP:name type are translated into HTTP header with corresponding 'name's.

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

- MCCHTTP.h

5.51 Arc::MCC_MsgValidator Class Reference

Inheritance diagram for Arc::MCC_MsgValidator:

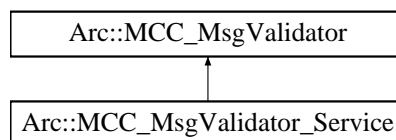


The documentation for this class was generated from the following file:

- MCCMsgValidator.h

5.52 Arc::MCC_MsgValidator_Service Class Reference

Inheritance diagram for Arc::MCC_MsgValidator_Service:



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

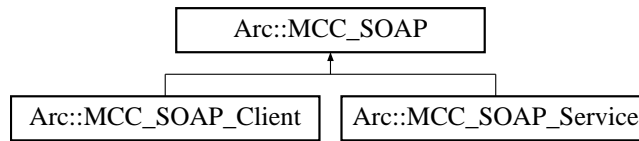
- MCCMsgValidator.h

5.53 Arc::MCC_SOAP Class Reference

A base class for SOAP client and service MCCs.

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

Inheritance diagram for Arc::MCC_SOAP:



5.53.1 Detailed Description

A base class for SOAP client and service MCCs.

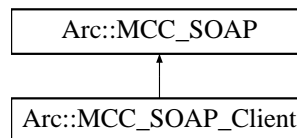
This is a base class for SOAP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

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

- MCCSOAP.h

5.54 Arc::MCC_SOAP_Client Class Reference

Inheritance diagram for Arc::MCC_SOAP_Client:



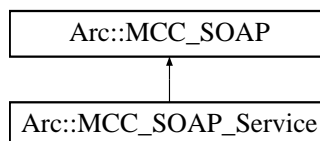
The documentation for this class was generated from the following file:

- MCCSOAP.h

5.55 Arc::MCC_SOAP_Service Class Reference

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

Inheritance diagram for Arc::MCC_SOAP_Service:



5.55.1 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:

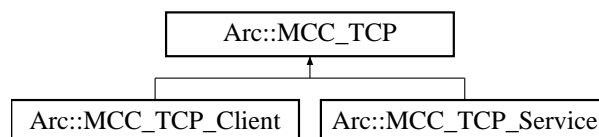
- MCCSOAP.h

5.56 Arc::MCC_TCP Class Reference

A base class for TCP client and service MCCs.

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

Inheritance diagram for Arc::MCC_TCP:



5.56.1 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.

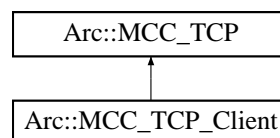
The documentation for this class was generated from the following file:

- MCCTCP.h

5.57 Arc::MCC_TCP_Client Class Reference

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

Inheritance diagram for Arc::MCC_TCP_Client:



5.57.1 Detailed Description

This class is MCC implementing TCP client. Upon creation it connects to specified TCP port 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.

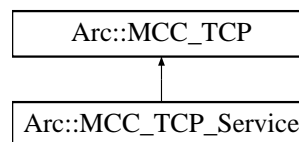
The documentation for this class was generated from the following file:

- MCCTCP.h

5.58 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`)

5.58.1 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`

5.58.2 Constructor & Destructor Documentation

5.58.2.1 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:

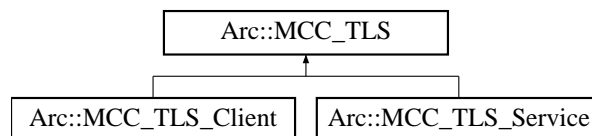
- MCCTCP.h

5.59 Arc::MCC_TLS Class Reference

A base class for TLS client and service MCCs.

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

Inheritance diagram for Arc::MCC_TLS:



5.59.1 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.

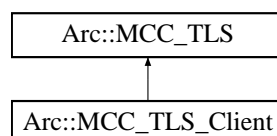
The documentation for this class was generated from the following file:

- MCCTLS.h

5.60 Arc::MCC_TLS_Client Class Reference

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

Inheritance diagram for Arc::MCC_TLS_Client:



5.60.1 Detailed Description

This class is MCC implementing TLS client.

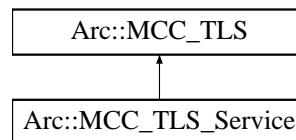
The documentation for this class was generated from the following file:

- MCCTLS.h

5.61 Arc::MCC_TLS_Service Class Reference

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

Inheritance diagram for Arc::MCC_TLS_Service:



5.61.1 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](#), which can create socket in the constructor method by using information in configuration file), we can only create "ssl" object which is bound to specified "socket", when [MCC_HTTP_Client](#) calls the process() method of [MCC_TLS_Client](#) object, or [MCC_TCP_Service](#) calls the process() method of [MCC_TLS_Service](#) object. The "ssl" object is embedded in a payload called PayloadTLSSocket.

The process() method of [MCC_TLS_Service](#) 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

5.62 Arc::PayloadGSISStream Class Reference

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

- [PayloadGSISStream.h](#)

5.63 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, bool head_response=false)
- [PayloadHTTP](#) (int code, const std::string &reason, bool head_response=false)
- 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_](#)

5.63.1 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.

5.63.2 Constructor & Destructor Documentation

5.63.2.1 `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.

5.63.2.2 `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.

5.63.2.3 `Arc::PayloadHTTP::PayloadHTTP (const std::string & method, const std::string & url)`

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

5.63.2.4 `Arc::PayloadHTTP::PayloadHTTP (int code, const std::string & reason, PayloadStreamInterface & stream, bool head_response = false)`

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

5.63.2.5 `Arc::PayloadHTTP::PayloadHTTP (int code, const std::string & reason, bool head_response = false)`

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

5.63.3 Member Function Documentation

5.63.3.1 `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.

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

Adds HTTP header attribute 'name' = 'value'

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

Returns all HTTP header attributes.

5.63.3.4 `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.

5.63.3.5 `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.

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

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

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

Read HTTP header and fill internal variables

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

Read up to 'size' bytes from stream_

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

Read from stream till

5.63.4 Field Documentation

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

true if connection should not be closed after response

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

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

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

Content-length of HTTP message

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

HTTP method being used or requested

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

true if content is chunked

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

HTTP reason being sent or supplied

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

minor number of HTTP version - must be 0 or 1

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

if true `stream_` is owned by this

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

HTTP code being sent or supplied

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

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

5.63.4.11 `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_`.

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

stream used to communicate to outside

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

if true `body_` is owned by this

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

URI being contacted

5.63.4.15 `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`

5.64 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)

5.64.1 Detailed Description

This class extends `PayloadStream` with TCP socket specific features

5.64.2 Constructor & Destructor Documentation

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

Constructor - connects to TCP server at specified hostname:port

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

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

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

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

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

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

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

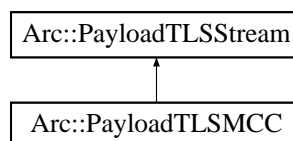
Copy constructor - inherits handle of copied object. Handle is NOT closed in destructor.

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

- PayloadTCPSocket.h

5.65 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)

5.65.1 Constructor & Destructor Documentation

5.65.1.1 `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.

5.65.1.2 `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.

5.65.1.3 `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.

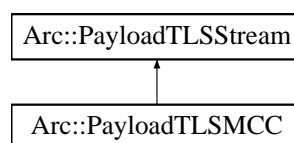
The documentation for this class was generated from the following file:

- `PayloadTLMCC.h`

5.66 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_](#)

5.66.1 Detailed Description

Implemetation of PayloadStreamInterface for SSL handle.

5.66.2 Constructor & Destructor Documentation

5.66.2.1 `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.

5.66.2.2 `virtual Arc::PayloadTLSStream::~~PayloadTLSStream (void)` [virtual]

Destructor.

5.66.3 Member Function Documentation

5.66.3.1 `X509* Arc::PayloadTLSStream::GetCert (void)`

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

5.66.3.2 `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.

5.66.3.3 `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.

5.66.4 Field Documentation

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

Timeout for read/write operations

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

- PayloadTLSStream.h

5.67 ArcSec::PDPServletInvoker Class Reference

[PDPServletInvoker](#) - client which will invoke pdpservice.

```
#include <PDPServletInvoker.h>
```

5.67.1 Detailed Description

[PDPServletInvoker](#) - client which will invoke pdpservice.

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

- PDPServletInvoker.h

5.68 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference

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

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

5.68.1 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

5.69 ArcSec::SAMLTokenSH Class Reference

Adds WS-Security SAML Token into SOAP Header.

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

5.69.1 Detailed Description

Adds WS-Security SAML Token into SOAP Header.

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

- SAMLTokenSH.h

5.70 ArcSec::SimpleListPDP Class Reference

Tests X509 subject against list of subjects in file.

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

5.70.1 Detailed Description

Tests X509 subject against list of subjects in file.

This class implements PDP interface. It's isPermitted() method compares X590 subject of requestor obtained from TLS layer (TLS:PEERDN) to list of subjects (ne per line) in external file. Locations of file is defined by 'location' attribute of PDP caonfiguration. Returns true if subject is present in list, otherwise false.

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

- SimpleListPDP.h

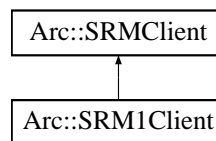
5.71 ArcSHCLegacy::SimpleMap Class Reference

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

- simplemap.h

5.72 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 [checkPermissions](#) (SRMClientRequest &)
- SRMReturnCode [getURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [getURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [putURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [putURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [releaseGet](#) (SRMClientRequest &req)
- SRMReturnCode [releasePut](#) (SRMClientRequest &req)
- SRMReturnCode [release](#) (SRMClientRequest &req)
- SRMReturnCode [abort](#) (SRMClientRequest &req)
- SRMReturnCode [info](#) (SRMClientRequest &req, std::list< struct [SRMFileMeta-Data](#) > &metadata, const int recursive=0, bool report_error=true)
- SRMReturnCode [remove](#) (SRMClientRequest &req)
- SRMReturnCode [copy](#) (SRMClientRequest &req, const std::string &source)

5.72.1 Member Function Documentation

5.72.1.1 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

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.2 SRMReturnCode Arc::SRM1Client::checkPermissions (SRMClientRequest & req) [inline, virtual]

Check permissions for the SURL in the request using the current credentials. req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

Copy a file between two SRM storages.

Parameters

<i>req</i>	The request object
<i>source</i>	The source SURL

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.4 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

<i>tokens</i>	The list filled by the service
<i>description</i>	The user request description, which can be specified when the request is created

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.5 `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

<i>tokens</i>	The list filled by the service
<i>description</i>	The space token description

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.6 `SRMReturnCode Arc::SRM1Client::getTURLs (SRMClientRequest & req, std::list< std::string > & urls) [virtual]`

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

Parameters

<i>req</i>	The request object
<i>urls</i>	A list of TURLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.7 `SRMReturnCode Arc::SRM1Client::getTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [inline, virtual]`

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

Parameters

<i>req</i>	The request object. Status must be ongoing.
<i>urls</i>	A list of TURLs filled by the method if the request completed successfully

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

```
5.72.1.8 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

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

Returns

SRMReturnCode specifying outcome of operation

See also

[SRMFileMetaData](#)

Implements [Arc::SRMClient](#).

```
5.72.1.9 SRMReturnCode Arc::SRM1Client::mkDir ( SRMClientRequest & req )
        [inline, virtual]
```

Make required directories for the SURL in the request

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

```
5.72.1.10 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

<i>version</i>	The version returned by the server
<i>report_error</i>	Whether an error should be reported

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.11 `SRMReturnCode Arc::SRM1Client::putURLs (SRMClientRequest & req, std::list< std::string > & urls) [virtual]`

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

Parameters

<i>req</i>	The request object
<i>urls</i>	A list of TURLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.12 `SRMReturnCode Arc::SRM1Client::putURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [inline, virtual]`

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

Parameters

<i>req</i>	The request object. Status must be ongoing.
<i>urls</i>	A list of TURLs filled by the method if the request completed successfully

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.13 `SRMReturnCode Arc::SRM1Client::release (SRMClientRequest & req)`
[virtual]

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

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.14 `SRMReturnCode Arc::SRM1Client::releaseGet (SRMClientRequest & req)`
[virtual]

Should be called after a successful copy from SRM storage.

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.15 `SRMReturnCode Arc::SRM1Client::releasePut (SRMClientRequest & req)`
[virtual]

Should be called after a successful copy to SRM storage.

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

5.72.1.16 `SRMReturnCode Arc::SRM1Client::remove (SRMClientRequest & req)`
[virtual]

Delete a file physically from storage and the SRM namespace.

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

Query the status of a request to bring files online. The SURLS map of the request object is updated if the status of any files in the request has changed. [requestBringOnline\(\)](#) but be called before this method.

Parameters

<i>req</i>	The request object to query the status of
------------	---

Returns

SRMReturnCode specifying outcome of operation

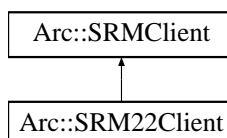
Implements [Arc::SRMClient](#).

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

- SRM1Client.h

5.73 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 [getURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [getURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [putURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [putURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [requestBringOnline](#) (SRMClientRequest &req)
- SRMReturnCode [requestBringOnlineStatus](#) (SRMClientRequest &req)
- SRMReturnCode [info](#) (SRMClientRequest &req, std::list< struct [SRMFileMeta-Data](#) > &metadata, 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)
- SRMReturnCode [checkPermissions](#) (SRMClientRequest &req)

5.73.1 Constructor & Destructor Documentation

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

Constructor

5.73.1.2 Arc::SRM22Client::~~SRM22Client ()

Destructor

5.73.2 Member Function Documentation

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

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

Implements [Arc::SRMClient](#).

5.73.2.2 SRMReturnCode Arc::SRM22Client::checkPermissions (SRMClientRequest & req)
[virtual]

Call srmCheckPermission

Implements [Arc::SRMClient](#).

5.73.2.3 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](#).

5.73.2.4 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](#).

5.73.2.5 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](#).

5.73.2.6 SRMReturnCode Arc::SRM22Client::getURLs (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) if the request is synchronous. If not it returns after making

the request. Although a list is returned, SRMv2.2 only returns one TURL per SURL.

Implements [Arc::SRMClient](#).

5.73.2.7 `SRMReturnCode Arc::SRM22Client::getTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [virtual]`

Uses `srmStatusOfGetRequest` to query the status of the given request.

Implements [Arc::SRMClient](#).

5.73.2.8 `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](#).

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

Call `srmMkDir`

Implements [Arc::SRMClient](#).

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

Get the server version from `srmPing`

Implements [Arc::SRMClient](#).

5.73.2.11 `SRMReturnCode Arc::SRM22Client::putTURLs (SRMClientRequest & req, std::list< std::string > & urls) [virtual]`

Retrieve TURLs which a file can be written to. Uses `srmPrepareToPut` and waits until a suitable TURL has been assigned if the request is synchronous. If not it returns after making the request. Although a list is returned, SRMv2.2 only returns one TURL per SURL.

Implements [Arc::SRMClient](#).

5.73.2.12 `SRMReturnCode Arc::SRM22Client::putTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [virtual]`

Uses `srmStatusOfPutRequest` to query the status of the given request.

Implements [Arc::SRMClient](#).

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

Not used in this version of SRM

Implements [Arc::SRMClient](#).

5.73.2.14 `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](#).

5.73.2.15 `SRMReturnCode Arc::SRM22Client::releasePut (SRMClientRequest & req)`
[virtual]

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

Implements [Arc::SRMClient](#).

5.73.2.16 `SRMReturnCode Arc::SRM22Client::remove (SRMClientRequest & req)`
[virtual]

Delete by srmRm or srmRmDir

Implements [Arc::SRMClient](#).

5.73.2.17 `SRMReturnCode Arc::SRM22Client::requestBringOnline (SRMClientRequest & req)` [virtual]

Call srmBringOnline with the URLs specified in req.

Implements [Arc::SRMClient](#).

5.73.2.18 `SRMReturnCode Arc::SRM22Client::requestBringOnlineStatus (SRMClientRequest & req)` [virtual]

Call srmStatusOfBringOnlineRequest and update req with any changes.

Implements [Arc::SRMClient](#).

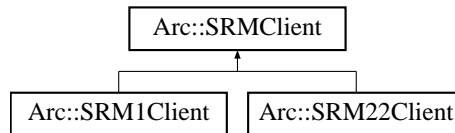
The documentation for this class was generated from the following file:

- SRM22Client.h

5.74 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 [getURLsStatus](#) (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)=0
- virtual SRMReturnCode [putURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode [releaseGet](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [releasePut](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [release](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [abort](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [info](#) (SRMClientRequest &req, std::list< struct [SRM-FileMetaData](#) > &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
- virtual SRMReturnCode [checkPermissions](#) (SRMClientRequest &req)=0

Static Public Member Functions

- static [SRMClient](#) * [getInstance](#) (const UserConfig &usercfg, const std::string &url, bool &timedout)

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 Logger [logger](#)

5.74.1 Detailed Description

A client interface to the SRM protocol. Instances of SRM clients are created by calling the [getInstance\(\)](#) 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.

5.74.2 Constructor & Destructor Documentation

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

Constructor

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

Destructor

5.74.3 Member Function Documentation

5.74.3.1 `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

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.2 `virtual SRMReturnCode Arc::SRMClient::checkPermissions (SRMClientRequest & req) [pure virtual]`

Check permissions for the SURL in the request using the current credentials. *req* The request object

Returns

SRMReturnCode specifying outcome of operation

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

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

Copy a file between two SRM storages.

Parameters

<i>req</i>	The request object
<i>source</i>	The source SURL

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.4 `static SRMClient* Arc::SRMClient::getInstance (const UserConfig & usercfg, const std::string & url, bool & timedout) [static]`

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

Parameters

<i>usercfg</i>	The user configuration.
<i>url</i>	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.

<i>timeout</i>	Whether the connection timed out
<i>conn_timeout</i>	Connection timeout to the SRM service

Returns

A pointer to an instance of [SRMClient](#) is returned, or NULL if it was not possible to create one.

5.74.3.5 `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

<i>tokens</i>	The list filled by the service
<i>description</i>	The user request description, which can be specified when the request is created

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.6 `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

<i>tokens</i>	The list filled by the service
<i>description</i>	The space token description

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.7 `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\(\)](#) is called to retrieve the transport URL(s) to copy the file from. It may be used synchronously or asynchronously, depending on the synchronous property of the request object. In the former case it will block until the TURLs are ready, in the latter case it will return after making the request and [getTURLsStatus\(\)](#) must be used to poll the request status if it was not completed.

Parameters

<i>req</i>	The request object
<i>urls</i>	A list of TURLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

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

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

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

Parameters

<i>req</i>	The request object. Status must be ongoing.
<i>urls</i>	A list of TURLs filled by the method if the request completed successfully

Returns

SRMReturnCode specifying outcome of operation

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

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

Returns the version of the SRM protocol used by this instance

References version.

5.74.3.10 `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

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

Returns

SRMReturnCode specifying outcome of operation

See also

[SRMFileMetaData](#)

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

5.74.3.11 `virtual SRMReturnCode Arc::SRMClient::mkDir (SRMClientRequest & req)`
`[pure virtual]`

Make required directories for the SURL in the request

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.12 `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 supresses the error message.

Parameters

<i>version</i>	The version returned by the server
<i>report_error</i>	Whether an error should be reported

Returns

SRMReturnCode specifying outcome of operation

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

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

Process SOAP request

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

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

Parameters

<i>req</i>	The request object
<i>urls</i>	A list of URLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

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

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

In the case where [putURLs](#) was called asynchronously and the request was not completed, this method should be called to poll the status of the request. [putURLs](#) must be called before this method and the request object must have ongoing request status.

Parameters

<i>req</i>	The request object. Status must be ongoing.
<i>urls</i>	A list of URLs filled by the method if the request completed successfully

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.16 `virtual SRMReturnCode Arc::SRMClient::release (SRMClientRequest & req)` [pure virtual]

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

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.17 `virtual SRMReturnCode Arc::SRMClient::releaseGet (SRMClientRequest & req)`
[pure virtual]

Should be called after a successful copy from SRM storage.

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.18 `virtual SRMReturnCode Arc::SRMClient::releasePut (SRMClientRequest & req)`
[pure virtual]

Should be called after a successful copy to SRM storage.

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.19 `virtual SRMReturnCode Arc::SRMClient::remove (SRMClientRequest & req)`
[pure virtual]

Delete a file physically from storage and the SRM namespace.

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

5.74.3.20 `virtual SRMReturnCode Arc::SRMClient::requestBringOnline (SRMClientRequest & req) [pure virtual]`

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

Parameters

<i>req</i>	The request object
------------	--------------------

Returns

SRMReturnCode specifying outcome of operation

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

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

Query the status of a request to bring files online. The SURIs map of the request object is updated if the status of any files in the request has changed. [requestBringOnline\(\)](#) but be called before this method.

Parameters

<i>req</i>	The request object to query the status of
------------	---

Returns

SRMReturnCode specifying outcome of operation

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

5.74.4 Field Documentation

5.74.4.1 `MCCConfig Arc::SRMClient::cfg [protected]`

SOAP configuraton object

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

SOAP client object

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

The implementation of the server

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

Logger

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

SOAP namespace

5.74.4.6 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.

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

Timeout for requests to the SRM service

5.74.4.8 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`

5.75 Arc::SRMClientRequest Class Reference

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

Public Member Functions

- [SRMClientRequest](#) (const std::list< std::string > &urls) throw (SRMInvalidRequestException)

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

5.75.1 Detailed Description

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

5.75.2 Constructor & Destructor Documentation

5.75.2.1 `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/

5.75.2.2 `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/

5.75.3 Member Function Documentation

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

set and get file id list

5.75.3.2 `void Arc::SRMClientRequest::finished_success () [inline]`

set and get status of request

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

set and get long list flag

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

set and get request id

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

set and get request timeout

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

set and get request token

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

set and get space token

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

set and get surl failures

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

set and get surl statuses

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

get URLs

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

set and get total size

5.75.3.12 void Arc::SRMClientRequest::transport_protocols (const std::list< std::string > &
protocols) [inline]

set and get transport protocols

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

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

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

- SRMClient.h

5.76 SRMFileInfo Class Reference

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

5.76.1 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

5.77 Arc::SRMFileMetaData Struct Reference

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

5.77.1 Detailed Description

File metadata

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

- SRMClient.h

5.78 SRMInfo Class Reference

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

5.78.1 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

5.79 Arc::SRMInvalidRequestException Class Reference

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

- SRMClient.h

5.80 SRMURL Class Reference

Public Member Functions

- [SRMURL](#) (std::string url)
- const std::string & [Endpoint](#) (void) const
- void [SetSRMVersion](#) (const std::string &version)
- 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](#) ()

5.80.1 Constructor & Destructor Documentation

5.80.1.1 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/

5.80.2 Member Function Documentation

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

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

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

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

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

eg /srm/managerv2

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

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

5.80.2.5 `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/dur`

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

Was the port number given in the constructor?

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

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

5.80.2.8 `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

5.81 ArcSHCLegacy::UnixMap Class Reference

Data Structures

- struct **source_t**
- class **unix_user_t**

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

- unixmap.h

5.82 ArcSec::UsernameTokenSH Class Reference

Adds WS-Security Username Token into SOAP Header.

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

5.82.1 Detailed Description

Adds WS-Security Username Token into SOAP Header.

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

- UsernameTokenSH.h

5.83 ArcSHCLegacy::voms Struct Reference

```
#include <auth.h>
```

Data Fields

- std::string [server](#)
- std::string [voname](#)
- std::vector< [voms_attrs](#) > [attrs](#)

5.83.1 Detailed Description

VOMS data

5.83.2 Field Documentation

5.83.2.1 std::vector<voms_attrs> ArcSHCLegacy::voms::attrs

User's characteristics

5.83.2.2 std::string ArcSHCLegacy::voms::server

The VOMS server DN, as from its certificate

5.83.2.3 std::string ArcSHCLegacy::voms::voname

The name of the VO to which the VOMS belongs

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

- [auth.h](#)

5.84 ArcSHCLegacy::voms_attrs Struct Reference

```
#include <auth.h>
```

Data Fields

- std::string [group](#)
- std::string [role](#)
- std::string [cap](#)

5.84.1 Detailed Description

VOMS attributes

5.84.2 Field Documentation

5.84.2.1 `std::string ArcSHCLegacy::voms_attrs::cap`

user's capability

5.84.2.2 `std::string ArcSHCLegacy::voms_attrs::group`

user's group

5.84.2.3 `std::string ArcSHCLegacy::voms_attrs::role`

user's role

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

- `auth.h`

5.85 ArcSec::X509TokenSH Class Reference

Adds WS-Security X509 Token into SOAP Header.

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

5.85.1 Detailed Description

Adds WS-Security X509 Token into SOAP Header.

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

- `X509TokenSH.h`

5.86 ArcSec::XACMLAlgFactory Class Reference

Algorithm factory class for XACML.

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

Public Member Functions

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

5.86.1 Detailed Description

Algorithm factory class for XACML.

5.86.2 Member Function Documentation

5.86.2.1 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](#) itself will release the Alg objects

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

- XACMLAlgFactory.h

5.87 ArcSec::XACMLApply Class Reference

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

- XACMLApply.h

5.88 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)

5.88.1 Detailed Description

Attribute factory class for XACML specified attributes.

5.88.2 Member Function Documentation

5.88.2.1 `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

5.89 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference

XACML specific AttributeProxy class.

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

Public Member Functions

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

5.89.1 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

5.90 ArcSec::XACMLCondition Class Reference

[XACMLCondition](#) class to parse and operate XACML specific <Condition> node.

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

Public Member Functions

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

5.90.1 Detailed Description

[XACMLCondition](#) class to parse and operate XACML specific <Condition> node.

5.90.2 Constructor & Destructor Documentation

5.90.2.1 **ArcSec::XACMLCondition::XACMLCondition** ([Arc::XMLNode](#) & *node*, [EvaluatorContext](#) * *ctx*)

Constructor -

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

- [XACMLCondition.h](#)

5.91 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)

5.91.1 Detailed Description

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

5.91.2 Constructor & Destructor Documentation

5.91.2.1 **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](#)

5.92 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)

5.92.1 Detailed Description

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

5.92.2 Member Function Documentation

5.92.2.1 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

5.93 ArcSec::XACMLFnFactory Class Reference

Function factory class for XACML specified attributes.

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

Public Member Functions

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

5.93.1 Detailed Description

Function factory class for XACML specified attributes.

5.93.2 Member Function Documentation

5.93.2.1 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](#) itself will release the Function objects

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

- XACMLFnFactory.h

5.94 ArcSec::XACMLPDP Class Reference

[XACMLPDP](#) - PDP which can handle the XACML specific request and policy schema.

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

5.94.1 Detailed Description

[XACMLPDP](#) - PDP which can handle the XACML specific request and policy schema.

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

- XACMLPDP.h

5.95 ArcSec::XACMLPolicy Class Reference

[XACMLPolicy](#) 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](#) ()

5.95.1 Detailed Description

[XACMLPolicy](#) class to parse and operate XACML specific <Policy> node.

5.95.2 Constructor & Destructor Documentation

5.95.2.1 ArcSec::XACMLPolicy::XACMLPolicy (void)

Constructor

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

Constructor

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

Constructor -

5.95.3 Member Function Documentation

5.95.3.1 `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

5.96 ArcSec::XACMLRequest Class Reference

Public Member Functions

- virtual const char * [getEvalName](#) () const
- virtual const char * [getName](#) () const

5.96.1 Member Function Documentation

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

Get the name of corresponding evaluator

5.96.1.2 `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

5.97 ArcSec::XACMLRule Class Reference

[XACMLRule](#) class to parse XACML specific <Rule> node.

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


5.97.1 Detailed Description

[XACMLRule](#) class to parse XACML specific <Rule> node.

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

- [XACMLRule.h](#)

5.98 ArcSec::XACMLTarget Class Reference

[XACMLTarget](#) class to parse and operate XACML specific <Target> node.

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

Public Member Functions

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

5.98.1 Detailed Description

[XACMLTarget](#) class to parse and operate XACML specific <Target> node.

5.98.2 Constructor & Destructor Documentation

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

Constructor -

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

- [XACMLTarget.h](#)

5.99 ArcSec::XACMLTargetMatch Class Reference

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

- [XACMLTarget.h](#)

5.100 ArcSec::XACMLTargetMatchGroup Class Reference

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

- [XACMLTarget.h](#)

5.101 ArcSec::XACMLTargetSection Class Reference

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

- XACMLTarget.h

Index

- ~LDAPQuery
 - Arc::LDAPQuery, [29](#)
- ~PayloadTLSStream
 - Arc::PayloadTLSStream, [46](#)
- ~SRM22Client
 - Arc::SRM22Client, [56](#)
- ~SRMClient
 - Arc::SRMClient, [61](#)
- abort
 - Arc::SRM1Client, [49](#)
 - Arc::SRM22Client, [57](#)
 - Arc::SRMClient, [61](#)
- AndList
 - ArcSec, [13](#)
- Arc::ConfigTLMCC, [23](#)
- Arc::DataPointARC, [23](#)
- Arc::DataPointFile, [24](#)
- Arc::DataPointGridFTP, [24](#)
- Arc::DataPointHTTP, [24](#)
- Arc::DataPointLDAP, [25](#)
- Arc::DataPointLFC, [25](#)
- Arc::DataPointRLS, [25](#)
- Arc::DataPointSRM, [26](#)
- Arc::DataPointXrootd, [26](#)
- Arc::LDAPQuery, [29](#)
 - ~LDAPQuery, [29](#)
 - LDAPQuery, [29](#)
 - Query, [29](#)
 - Result, [29](#)
- Arc::Lister, [30](#)
- Arc::MCC_GSI_Client, [31](#)
- Arc::MCC_GSI_Service, [31](#)
- Arc::MCC_HTTP, [31](#)
- Arc::MCC_HTTP_Client, [31](#)
- Arc::MCC_HTTP_Service, [32](#)
- Arc::MCC_MsgValidator, [33](#)
- Arc::MCC_MsgValidator_Service, [33](#)
- Arc::MCC_SOAP, [33](#)
- Arc::MCC_SOAP_Client, [34](#)
- Arc::MCC_SOAP_Service, [34](#)
- Arc::MCC_TCP, [35](#)
- Arc::MCC_TCP_Client, [35](#)
- Arc::MCC_TCP_Service, [36](#)
 - MCC_TCP_Service, [37](#)
- Arc::MCC_TLS, [37](#)
- Arc::MCC_TLS_Client, [37](#)
- Arc::MCC_TLS_Service, [38](#)
- Arc::PayloadGSISStream, [38](#)
- Arc::PayloadHTTP, [39](#)
 - Attribute, [40](#)
 - Attributes, [41](#)
 - attributes_, [41](#)
 - Body, [41](#)
 - body_own_, [42](#)
 - chunked_, [42](#)
 - code_, [42](#)
 - Flush, [41](#)
 - get_body, [41](#)
 - keep_alive_, [42](#)
 - length_, [42](#)
 - method_, [42](#)
 - parse_header, [41](#)
 - PayloadHTTP, [40](#)
 - rbody_, [42](#)
 - read, [41](#)
 - readline, [41](#)
 - reason_, [42](#)
 - sbody_, [42](#)
 - stream_, [42](#)
 - stream_own_, [43](#)
 - uri_, [43](#)
 - version_major_, [43](#)
 - version_minor_, [43](#)
- Arc::PayloadTCPSocket, [43](#)
 - PayloadTCPSocket, [44](#)
- Arc::PayloadTLMCC, [44](#)
 - PayloadTLMCC, [45](#)
- Arc::PayloadTLSStream, [45](#)
 - ~PayloadTLSStream, [46](#)
 - GetCert, [46](#)
 - GetPeerCert, [46](#)

- PayloadTLSStream, 46
- ssl_, 47
- STACK_OF, 46
- Arc::SRM1Client, 48
 - abort, 49
 - checkPermissions, 49
 - copy, 50
 - getRequestTokens, 50
 - getSpaceTokens, 50
 - getURLs, 51
 - getURLsStatus, 51
 - info, 52
 - mkDir, 52
 - ping, 52
 - putURLs, 53
 - putURLsStatus, 53
 - release, 53
 - releaseGet, 54
 - releasePut, 54
 - remove, 54
 - requestBringOnline, 55
 - requestBringOnlineStatus, 55
- Arc::SRM22Client, 56
 - ~SRM22Client, 56
 - abort, 57
 - checkPermissions, 57
 - copy, 57
 - getRequestTokens, 57
 - getSpaceTokens, 57
 - getURLs, 57
 - getURLsStatus, 58
 - info, 58
 - mkDir, 58
 - ping, 58
 - putURLs, 58
 - putURLsStatus, 58
 - release, 59
 - releaseGet, 59
 - releasePut, 59
 - remove, 59
 - requestBringOnline, 59
 - requestBringOnlineStatus, 59
 - SRM22Client, 56
- Arc::SRMClient, 60
 - ~SRMClient, 61
 - abort, 61
 - cfg, 68
 - checkPermissions, 62
 - client, 68
 - copy, 62
 - getInstance, 62
 - getRequestTokens, 63
 - getSpaceTokens, 63
 - getURLs, 63
 - getURLsStatus, 64
 - getVersion, 64
 - implementation, 69
 - info, 64
 - logger, 69
 - mkDir, 65
 - ns, 69
 - ping, 65
 - process, 65
 - putURLs, 66
 - putURLsStatus, 66
 - release, 66
 - releaseGet, 67
 - releasePut, 67
 - remove, 67
 - requestBringOnline, 68
 - requestBringOnlineStatus, 68
 - service_endpoint, 69
 - SRMClient, 61
 - user_timeout, 69
 - version, 69
- Arc::SRMClientRequest, 69
 - file_ids, 70
 - finished_success, 70
 - long_list, 70
 - request_id, 71
 - request_timeout, 71
 - request_token, 71
 - space_token, 71
 - SRMClientRequest, 70
 - surl_failures, 71
 - surl_statuses, 71
 - surls, 71
 - total_size, 71
 - transport_protocols, 71
 - waiting_time, 71
- Arc::SRMFileMetaData, 72
- Arc::SRMInvalidRequestException, 73
- ArcEvaluationCtx
 - ArcSec::ArcEvaluationCtx, 18
- ArcPolicy
 - ArcSec::ArcPolicy, 21
- ArcSec, 11
 - AndList, 13
 - Match, 13
- ArcSec::AllowPDP, 15

- ArcSec::ArcAlgFactory, 15
 - createAlg, 16
- ArcSec::ArcAttributeFactory, 16
 - createValue, 16
- ArcSec::ArcAttributeProxy, 16
- ArcSec::ArcAuthZ, 17
 - Handle, 18
 - MakePDPs, 18
- ArcSec::ArcEvaluationCtx, 18
 - ArcEvaluationCtx, 18
 - split, 18
- ArcSec::ArcEvaluator, 19
 - evaluate, 19
- ArcSec::ArcFnFactory, 19
 - createFn, 20
- ArcSec::ArcPDP, 20
- ArcSec::ArcPolicy, 20
 - ArcPolicy, 21
 - make_policy, 21
- ArcSec::ArcRequest, 21
- ArcSec::ArcRequestItem, 21
- ArcSec::ArcRequestTuple, 22
- ArcSec::ArcRule, 22
- ArcSec::AttributeDesignator, 22
- ArcSec::AttributeSelector, 22
- ArcSec::DelegationCollector, 27
- ArcSec::DelegationMultiSecAttr, 27
- ArcSec::DelegationPDP, 27
- ArcSec::DelegationSecAttr, 27
- ArcSec::DelegationSH, 27
- ArcSec::DenyPDP, 27
- ArcSec::GACLEvaluator, 28
 - evaluate, 28
- ArcSec::GACLPDP, 28
- ArcSec::GACLPolicy, 28
- ArcSec::GACLRequest, 28
- ArcSec::PDPServiceInvoker, 47
- ArcSec::SAML2SSO_AssertionConsumerSHAttribute, 47
- ArcSec::SAMLTokenSH, 47
- ArcSec::SimpleListPDP, 48
- ArcSec::UsernameTokenSH, 74
- ArcSec::X509TokenSH, 76
- ArcSec::XACMLAlgFactory, 76
 - createAlg, 77
- ArcSec::XACMLApply, 77
- ArcSec::XACMLAttributeFactory, 77
 - createValue, 78
- ArcSec::XACMLAttributeProxy, 78
- ArcSec::XACMLCondition, 78
 - XACMLCondition, 79
- ArcSec::XACMLEvaluationCtx, 79
 - XACMLEvaluationCtx, 79
- ArcSec::XACMLEvaluator, 79
 - evaluate, 80
- ArcSec::XACMLFnFactory, 80
 - createFn, 80
- ArcSec::XACMLPDP, 81
- ArcSec::XACMLPolicy, 81
 - make_policy, 82
 - XACMLPolicy, 81
- ArcSec::XACMLRequest, 82
 - getEvalName, 82
 - getName, 82
- ArcSec::XACMLRule, 82
- ArcSec::XACMLTarget, 83
 - XACMLTarget, 83
- ArcSec::XACMLTargetMatch, 83
- ArcSec::XACMLTargetMatchGroup, 83
- ArcSec::XACMLTargetSection, 84
- ArcSHCLegacy::AuthUser, 22
- ArcSHCLegacy::AuthVO, 23
- ArcSHCLegacy::ConfigParser, 23
- ArcSHCLegacy::LegacyMap, 30
- ArcSHCLegacy::LegacyPDP, 30
- ArcSHCLegacy::LegacySecAttr, 30
- ArcSHCLegacy::LegacySecHandler, 30
- ArcSHCLegacy::SimpleMap, 48
- ArcSHCLegacy::UnixMap, 74
- ArcSHCLegacy::voms, 75
 - attrs, 75
 - server, 75
 - voname, 75
- ArcSHCLegacy::voms_attrs, 75
 - cap, 76
 - group, 76
 - role, 76
- Attribute
 - Arc::PayloadHTTP, 40
- Attributes
 - Arc::PayloadHTTP, 41
- attributes_
 - Arc::PayloadHTTP, 41
- attrs
 - ArcSHCLegacy::voms, 75
- BaseURL
 - SRMURL, 73
- Body
 - Arc::PayloadHTTP, 41

- body_own_
 - Arc::PayloadHTTP, 42
- cap
 - ArcSHCLegacy::voms_attrs, 76
- cfg
 - Arc::SRMClient, 68
- checkPermissions
 - Arc::SRM1Client, 49
 - Arc::SRM22Client, 57
 - Arc::SRMClient, 62
- chunked_
 - Arc::PayloadHTTP, 42
- client
 - Arc::SRMClient, 68
- code_
 - Arc::PayloadHTTP, 42
- ContactURL
 - SRMURL, 73
- copy
 - Arc::SRM1Client, 50
 - Arc::SRM22Client, 57
 - Arc::SRMClient, 62
- createAlg
 - ArcSec::ArcAlgFactory, 16
 - ArcSec::XACMLAlgFactory, 77
- createFn
 - ArcSec::ArcFnFactory, 20
 - ArcSec::XACMLFnFactory, 80
- createValue
 - ArcSec::ArcAttributeFactory, 16
 - ArcSec::XACMLAttributeFactory, 78
- Endpoint
 - SRMURL, 73
- evaluate
 - ArcSec::ArcEvaluator, 19
 - ArcSec::GACLEvaluator, 28
 - ArcSec::XACMLEvaluator, 80
- file_ids
 - Arc::SRMClientRequest, 70
- FileName
 - SRMURL, 73
- finished_success
 - Arc::SRMClientRequest, 70
- Flush
 - Arc::PayloadHTTP, 41
- FullURL
 - SRMURL, 73
- get_body
 - Arc::PayloadHTTP, 41
- GetCert
 - Arc::PayloadTLSStream, 46
- getEvalName
 - ArcSec::XACMLRequest, 82
- getInstance
 - Arc::SRMClient, 62
- getName
 - ArcSec::XACMLRequest, 82
- GetPeerCert
 - Arc::PayloadTLSStream, 46
- getRequestTokens
 - Arc::SRM1Client, 50
 - Arc::SRM22Client, 57
 - Arc::SRMClient, 63
- getSpaceTokens
 - Arc::SRM1Client, 50
 - Arc::SRM22Client, 57
 - Arc::SRMClient, 63
- getURLs
 - Arc::SRM1Client, 51
 - Arc::SRM22Client, 57
 - Arc::SRMClient, 63
- getURLsStatus
 - Arc::SRM1Client, 51
 - Arc::SRM22Client, 58
 - Arc::SRMClient, 64
- getVersion
 - Arc::SRMClient, 64
- group
 - ArcSHCLegacy::voms_attrs, 76
- Handle
 - ArcSec::ArcAuthZ, 18
- implementation
 - Arc::SRMClient, 69
- info
 - Arc::SRM1Client, 52
 - Arc::SRM22Client, 58
 - Arc::SRMClient, 64
- keep_alive_
 - Arc::PayloadHTTP, 42
- LDAPQuery
 - Arc::LDAPQuery, 29
- length_
 - Arc::PayloadHTTP, 42

- logger
 - Arc::SRMClient, 69
- long_list
 - Arc::SRMClientRequest, 70
- make_policy
 - ArcSec::ArcPolicy, 21
 - ArcSec::XACMLPolicy, 82
- MakePDPs
 - ArcSec::ArcAuthZ, 18
- Match
 - ArcSec, 13
- MCC_TCP_Service
 - Arc::MCC_TCP_Service, 37
- method_
 - Arc::PayloadHTTP, 42
- mkdir
 - Arc::SRM1Client, 52
 - Arc::SRM22Client, 58
 - Arc::SRMClient, 65
- ns
 - Arc::SRMClient, 69
- parse_header
 - Arc::PayloadHTTP, 41
- PayloadHTTP
 - Arc::PayloadHTTP, 40
- PayloadTCPSocket
 - Arc::PayloadTCPSocket, 44
- PayloadTLMCC
 - Arc::PayloadTLMCC, 45
- PayloadTLSSStream
 - Arc::PayloadTLSSStream, 46
- ping
 - Arc::SRM1Client, 52
 - Arc::SRM22Client, 58
 - Arc::SRMClient, 65
- PortDefined
 - SRMURL, 74
- process
 - Arc::SRMClient, 65
- putURLs
 - Arc::SRM1Client, 53
 - Arc::SRM22Client, 58
 - Arc::SRMClient, 66
- putURLsStatus
 - Arc::SRM1Client, 53
 - Arc::SRM22Client, 58
 - Arc::SRMClient, 66
- Query
 - Arc::LDAPQuery, 29
- rbody_
 - Arc::PayloadHTTP, 42
- read
 - Arc::PayloadHTTP, 41
- readline
 - Arc::PayloadHTTP, 41
- reason_
 - Arc::PayloadHTTP, 42
- release
 - Arc::SRM1Client, 53
 - Arc::SRM22Client, 59
 - Arc::SRMClient, 66
- releaseGet
 - Arc::SRM1Client, 54
 - Arc::SRM22Client, 59
 - Arc::SRMClient, 67
- releasePut
 - Arc::SRM1Client, 54
 - Arc::SRM22Client, 59
 - Arc::SRMClient, 67
- remove
 - Arc::SRM1Client, 54
 - Arc::SRM22Client, 59
 - Arc::SRMClient, 67
- request_id
 - Arc::SRMClientRequest, 71
- request_timeout
 - Arc::SRMClientRequest, 71
- request_token
 - Arc::SRMClientRequest, 71
- requestBringOnline
 - Arc::SRM1Client, 55
 - Arc::SRM22Client, 59
 - Arc::SRMClient, 68
- requestBringOnlineStatus
 - Arc::SRM1Client, 55
 - Arc::SRM22Client, 59
 - Arc::SRMClient, 68
- Result
 - Arc::LDAPQuery, 29
- role
 - ArcSHCLegacy::voms_attrs, 76
- sbody_
 - Arc::PayloadHTTP, 42
- server
 - ArcSHCLegacy::voms, 75

- service_endpoint
 - Arc::SRMClient, 69
- SetSRMVersion
 - SRMURL, 74
- ShortURL
 - SRMURL, 74
- space_token
 - Arc::SRMClientRequest, 71
- split
 - ArcSec::ArcEvaluationCtx, 18
- SRM22Client
 - Arc::SRM22Client, 56
- SRMClient
 - Arc::SRMClient, 61
- SRMClientRequest
 - Arc::SRMClientRequest, 70
- SRMFileInfo, 72
- SRMInfo, 72
- SRMURL, 73
 - BaseURL, 73
 - ContactURL, 73
 - Endpoint, 73
 - FileName, 73
 - FullURL, 73
 - PortDefined, 74
 - SetSRMVersion, 74
 - ShortURL, 74
 - SRMURL, 73
- ssl_
 - Arc::PayloadTLSStream, 47
- STACK_OF
 - Arc::PayloadTLSStream, 46
- stream_
 - Arc::PayloadHTTP, 42
- stream_own_
 - Arc::PayloadHTTP, 43
- surl_failures
 - Arc::SRMClientRequest, 71
- surl_statuses
 - Arc::SRMClientRequest, 71
- surls
 - Arc::SRMClientRequest, 71
- total_size
 - Arc::SRMClientRequest, 71
- transport_protocols
 - Arc::SRMClientRequest, 71
- uri_
 - Arc::PayloadHTTP, 43
- user_timeout
 - Arc::SRMClient, 69
- version
 - Arc::SRMClient, 69
- version_major_
 - Arc::PayloadHTTP, 43
- version_minor_
 - Arc::PayloadHTTP, 43
- voname
 - ArcSHCLegacy::voms, 75
- waiting_time
 - Arc::SRMClientRequest, 71
- XACMLCondition
 - ArcSec::XACMLCondition, 79
- XACMLEvaluationCtx
 - ArcSec::XACMLEvaluationCtx, 79
- XACMLPolicy
 - ArcSec::XACMLPolicy, 81
- XACMLTarget
 - ArcSec::XACMLTarget, 83
