

Hosting Environment (Daemon) Chain Components

Generated by Doxygen 1.7.1

Tue Dec 21 2010 22:35:58

Contents

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

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

5.16	ArcSec::AttributeSelector Class Reference	22
5.17	Arc::ConfigTLSMCC Class Reference	22
5.18	Arc::DataPointARC Class Reference	22
5.19	Arc::DataPointFile Class Reference	22
5.20	Arc::DataPointGridFTP Class Reference	22
5.21	Arc::DataPointHTTP Class Reference	22
5.22	Arc::DataPointLDAP Class Reference	22
5.23	Arc::DataPointLFC Class Reference	23
5.24	Arc::DataPointRLS Class Reference	23
5.25	Arc::DataPointSRM Class Reference	23
5.26	ArcSec::DelegationCollector Class Reference	23
5.27	ArcSec::DelegationMultiSecAttr Class Reference	23
5.28	ArcSec::DelegationPDP Class Reference	23
5.28.1	Detailed Description	23
5.29	ArcSec::DelegationSecAttr Class Reference	24
5.30	ArcSec::DelegationSH Class Reference	24
5.31	ArcSec::DenyPDP Class Reference	24
5.31.1	Detailed Description	24
5.32	ArcSec::GACLEvaluator Class Reference	24
5.32.1	Member Function Documentation	24
5.32.1.1	evaluate	24
5.33	ArcSec::GACLPDP Class Reference	24
5.34	ArcSec::GACLPolicy Class Reference	25
5.35	ArcSec::GACLRequest Class Reference	25
5.36	Arc::HTTPResponseHeader Class Reference	25
5.37	Arc::HTTPSClient Class Reference	25
5.37.1	Member Function Documentation	25
5.37.1.1	connect	25
5.38	Arc::HTTPSClientConnector Class Reference	25
5.38.1	Member Function Documentation	26
5.38.1.1	transfer	26
5.39	Arc::HTTPSClientConnectorGlobus Class Reference	26
5.39.1	Member Function Documentation	27
5.39.1.1	transfer	27
5.40	Arc::HTTPSClientConnectorGSSAPI Class Reference	27
5.40.1	Member Function Documentation	28

5.40.1.1	transfer	28
5.41	Arc::HTTPSClientSOAP Class Reference	28
5.42	Arc::LDAPQuery Class Reference	28
5.42.1	Detailed Description	28
5.42.2	Constructor & Destructor Documentation	29
5.42.2.1	LDAPQuery	29
5.42.2.2	~LDAPQuery	29
5.42.3	Member Function Documentation	29
5.42.3.1	Query	29
5.42.3.2	Result	29
5.43	Arc::Lister Class Reference	29
5.44	Arc::MCC_GSI_Client Class Reference	29
5.45	Arc::MCC_GSI_Service Class Reference	29
5.46	Arc::MCC_HTTP Class Reference	30
5.46.1	Detailed Description	30
5.47	Arc::MCC_HTTP_Client Class Reference	30
5.47.1	Detailed Description	30
5.48	Arc::MCC_HTTP_Service Class Reference	31
5.48.1	Detailed Description	31
5.49	Arc::MCC_SOAP Class Reference	31
5.49.1	Detailed Description	31
5.50	Arc::MCC_SOAP_Client Class Reference	32
5.51	Arc::MCC_SOAP_Service Class Reference	32
5.51.1	Detailed Description	32
5.52	Arc::MCC_TCP Class Reference	32
5.52.1	Detailed Description	33
5.53	Arc::MCC_TCP_Client Class Reference	33
5.53.1	Detailed Description	33
5.54	Arc::MCC_TCP_Service Class Reference	33
5.54.1	Detailed Description	34
5.54.2	Constructor & Destructor Documentation	34
5.54.2.1	MCC_TCP_Service	34
5.55	Arc::MCC_TLS Class Reference	34
5.55.1	Detailed Description	34
5.56	Arc::MCC_TLS_Client Class Reference	35
5.56.1	Detailed Description	35

5.57 Arc::MCC_TLS_Service Class Reference	35
5.57.1 Detailed Description	35
5.58 Arc::PayloadGSISStream Class Reference	36
5.59 Arc::PayloadHTTP Class Reference	36
5.59.1 Detailed Description	37
5.59.2 Constructor & Destructor Documentation	37
5.59.2.1 PayloadHTTP	37
5.59.2.2 PayloadHTTP	37
5.59.2.3 PayloadHTTP	37
5.59.2.4 PayloadHTTP	37
5.59.2.5 PayloadHTTP	37
5.59.3 Member Function Documentation	37
5.59.3.1 Attribute	37
5.59.3.2 Attribute	37
5.59.3.3 Attributes	38
5.59.3.4 Body	38
5.59.3.5 Flush	38
5.59.3.6 get_body	38
5.59.3.7 parse_header	38
5.59.3.8 read	38
5.59.3.9 readline	38
5.59.4 Field Documentation	38
5.59.4.1 attributes_	38
5.59.4.2 body_own_	38
5.59.4.3 chunked_	38
5.59.4.4 code_	39
5.59.4.5 keep_alive_	39
5.59.4.6 length_	39
5.59.4.7 method_	39
5.59.4.8 rbody_	39
5.59.4.9 reason_	39
5.59.4.10 sbody_	39
5.59.4.11 stream_	39
5.59.4.12 stream_own_	39
5.59.4.13 uri_	39
5.59.4.14 version_major_	39

5.59.4.15 version_minor_	40
5.60 Arc::PayloadTCPSocket Class Reference	40
5.60.1 Detailed Description	40
5.60.2 Constructor & Destructor Documentation	40
5.60.2.1 PayloadTCPSocket	40
5.60.2.2 PayloadTCPSocket	40
5.60.2.3 PayloadTCPSocket	40
5.60.2.4 PayloadTCPSocket	40
5.60.2.5 PayloadTCPSocket	41
5.61 Arc::PayloadTLMCC Class Reference	41
5.61.1 Constructor & Destructor Documentation	41
5.61.1.1 PayloadTLMCC	41
5.61.1.2 PayloadTLMCC	41
5.61.1.3 PayloadTLMCC	41
5.62 Arc::PayloadTLSSStream Class Reference	42
5.62.1 Detailed Description	42
5.62.2 Constructor & Destructor Documentation	42
5.62.2.1 PayloadTLSSStream	42
5.62.2.2 ~PayloadTLSSStream	42
5.62.3 Member Function Documentation	42
5.62.3.1 GetCert	42
5.62.3.2 GetPeerCert	43
5.62.3.3 STACK_OF	43
5.62.4 Field Documentation	43
5.62.4.1 ssl_	43
5.63 ArcSec::PDPSERVICEInvoker Class Reference	43
5.63.1 Detailed Description	43
5.64 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference	43
5.64.1 Detailed Description	43
5.65 ArcSec::SAMLTokenSH Class Reference	44
5.65.1 Detailed Description	44
5.66 ArcSec::SimpleListPDP Class Reference	44
5.66.1 Detailed Description	44
5.67 SRM1Client Class Reference	44
5.67.1 Member Function Documentation	45
5.67.1.1 abort	45

5.67.1.2	copy	45
5.67.1.3	getRequestTokens	45
5.67.1.4	getSpaceTokens	46
5.67.1.5	getTURLs	46
5.67.1.6	info	46
5.67.1.7	mkDir	47
5.67.1.8	ping	47
5.67.1.9	putTURLs	48
5.67.1.10	release	48
5.67.1.11	releaseGet	48
5.67.1.12	releasePut	48
5.67.1.13	remove	49
5.67.1.14	requestBringOnline	49
5.67.1.15	requestBringOnlineStatus	49
5.68	SRM22Client Class Reference	50
5.68.1	Member Function Documentation	50
5.68.1.1	abort	50
5.68.1.2	copy	50
5.68.1.3	getRequestTokens	50
5.68.1.4	getSpaceTokens	51
5.68.1.5	getTURLs	51
5.68.1.6	info	51
5.68.1.7	mkDir	51
5.68.1.8	ping	51
5.68.1.9	putTURLs	51
5.68.1.10	release	51
5.68.1.11	releaseGet	52
5.68.1.12	releasePut	52
5.68.1.13	remove	52
5.68.1.14	requestBringOnline	52
5.68.1.15	requestBringOnlineStatus	52
5.69	SRMClient Class Reference	52
5.69.1	Detailed Description	53
5.69.2	Constructor & Destructor Documentation	54
5.69.2.1	~SRMClient	54
5.69.3	Member Function Documentation	54

5.69.3.1	abort	54
5.69.3.2	connect	54
5.69.3.3	copy	54
5.69.3.4	disconnect	54
5.69.3.5	getInstance	55
5.69.3.6	getRequestTokens	55
5.69.3.7	getSpaceTokens	55
5.69.3.8	getTURLs	56
5.69.3.9	getVersion	56
5.69.3.10	info	56
5.69.3.11	mkDir	56
5.69.3.12	ping	57
5.69.3.13	putTURLs	57
5.69.3.14	release	57
5.69.3.15	releaseGet	58
5.69.3.16	releasePut	58
5.69.3.17	remove	58
5.69.3.18	requestBringOnline	59
5.69.3.19	requestBringOnlineStatus	59
5.69.3.20	Timeout	59
5.69.4	Field Documentation	59
5.69.4.1	csoap	59
5.69.4.2	implementation	59
5.69.4.3	logger	59
5.69.4.4	request_timeout	60
5.69.4.5	service_endpoint	60
5.69.4.6	version	60
5.70	SRMClientRequest Class Reference	60
5.70.1	Detailed Description	60
5.70.2	Constructor & Destructor Documentation	61
5.70.2.1	SRMClientRequest	61
5.70.2.2	SRMClientRequest	61
5.70.3	Member Function Documentation	61
5.70.3.1	file_ids	61
5.70.3.2	finished_success	61
5.70.3.3	long_list	61

5.70.3.4	request_id	61
5.70.3.5	request_token	61
5.70.3.6	space_token	61
5.70.3.7	surl_failures	61
5.70.3.8	surl_statuses	62
5.70.3.9	surls	62
5.70.3.10	waiting_time	62
5.71	SRMFileInfo Class Reference	62
5.71.1	Detailed Description	62
5.72	SRMFileMetaData Struct Reference	62
5.72.1	Detailed Description	62
5.73	SRMInfo Class Reference	62
5.73.1	Detailed Description	63
5.74	SRMInvalidRequestException Class Reference	63
5.75	SRMURL Class Reference	63
5.75.1	Constructor & Destructor Documentation	63
5.75.1.1	SRMURL	63
5.75.2	Member Function Documentation	63
5.75.2.1	BaseURL	63
5.75.2.2	ContactURL	63
5.75.2.3	Endpoint	63
5.75.2.4	FileName	64
5.75.2.5	FullURL	64
5.75.2.6	PortDefined	64
5.75.2.7	SetSRMVersion	64
5.75.2.8	ShortURL	64
5.76	ArcSec::UsernameTokenSH Class Reference	64
5.76.1	Detailed Description	64
5.77	ArcSec::X509TokenSH Class Reference	64
5.77.1	Detailed Description	65
5.78	ArcSec::XACMLAlgFactory Class Reference	65
5.78.1	Detailed Description	65
5.78.2	Member Function Documentation	65
5.78.2.1	createAlg	65
5.79	ArcSec::XACMLApply Class Reference	65
5.80	ArcSec::XACMLAttributeFactory Class Reference	65

5.80.1 Detailed Description	66
5.80.2 Member Function Documentation	66
5.80.2.1 createValue	66
5.81 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference	66
5.81.1 Detailed Description	66
5.82 ArcSec::XACMLCondition Class Reference	66
5.82.1 Detailed Description	67
5.82.2 Constructor & Destructor Documentation	67
5.82.2.1 XACMLCondition	67
5.83 ArcSec::XACMLEvaluationCtx Class Reference	67
5.83.1 Detailed Description	67
5.83.2 Constructor & Destructor Documentation	67
5.83.2.1 XACMLEvaluationCtx	67
5.84 ArcSec::XACMLEvaluator Class Reference	67
5.84.1 Detailed Description	68
5.84.2 Member Function Documentation	68
5.84.2.1 evaluate	68
5.85 ArcSec::XACMLFnFactory Class Reference	68
5.85.1 Detailed Description	68
5.85.2 Member Function Documentation	68
5.85.2.1 createFn	68
5.86 ArcSec::XACMLPDP Class Reference	69
5.86.1 Detailed Description	69
5.87 ArcSec::XACMLPolicy Class Reference	69
5.87.1 Detailed Description	69
5.87.2 Constructor & Destructor Documentation	69
5.87.2.1 XACMLPolicy	69
5.87.2.2 XACMLPolicy	69
5.87.2.3 XACMLPolicy	69
5.87.3 Member Function Documentation	70
5.87.3.1 make_policy	70
5.88 ArcSec::XACMLRequest Class Reference	70
5.88.1 Member Function Documentation	70
5.88.1.1 getEvalName	70
5.88.1.2 getName	70
5.89 ArcSec::XACMLRule Class Reference	70

5.89.1 Detailed Description	70
5.90 ArcSec::XACMLTarget Class Reference	71
5.90.1 Detailed Description	71
5.90.2 Constructor & Destructor Documentation	71
5.90.2.1 XACMLTarget	71
5.91 ArcSec::XACMLTargetMatch Class Reference	71
5.92 ArcSec::XACMLTargetMatchGroup Class Reference	71
5.93 ArcSec::XACMLTargetSection Class Reference	71

Chapter 1

Namespace Index

1.1 Namespace List

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

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

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

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

ArcSec::GACLPolicy	25
ArcSec::GACLRequest	25
Arc::HTTPResponseHeader	25
Arc::HTTPSCClient	25
Arc::HTTPSCClientSOAP	28
Arc::HTTPSCClientConnector	25
Arc::HTTPSCClientConnectorGlobus	26
Arc::HTTPSCClientConnectorGSSAPI	27
Arc::LDAPQuery	28
Arc::Lister	29
Arc::MCC_GSI_Client	29
Arc::MCC_GSI_Service	29
Arc::MCC_HTTP	30
Arc::MCC_HTTP_Client	30
Arc::MCC_HTTP_Service	31
Arc::MCC_SOAP	31
Arc::MCC_SOAP_Client	32
Arc::MCC_SOAP_Service	32
Arc::MCC_TCP	32
Arc::MCC_TCP_Client	33
Arc::MCC_TCP_Service	33
Arc::MCC_TLS	34
Arc::MCC_TLS_Client	35
Arc::MCC_TLS_Service	35
Arc::PayloadGSISStream	36
Arc::PayloadHTTP	36
Arc::PayloadTCPSToken	40
Arc::PayloadTLSSStream	42
Arc::PayloadTLSTMCC	41
ArcSec::PDPServletInvoker	43
ArcSec::SAML2SSO_AssertionConsumerSH	43
ArcSec::SAMLTokenSH	44
ArcSec::SimpleListPDP	44
SRMClient	52
SRM1Client	44
SRM22Client	50
SRMClientRequest	60
SRMFileInfo	62
SRMFileMetaData	62
SRMInfo	62
SRMInvalidRequestException	63
SRMURL	63
ArcSec::UsernameTokenSH	64
ArcSec::X509TokenSH	64
ArcSec::XACMLAlgFactory	65
ArcSec::XACMLApply	65
ArcSec::XACMLAttributeFactory	65
ArcSec::XACMLAttributeProxy< TheAttribute >	66
ArcSec::XACMLCondition	66
ArcSec::XACMLEvaluationCtx	67
ArcSec::XACMLEvaluator	67

ArcSec::XACMLFnFactory	68
ArcSec::XACMLPDP	69
ArcSec::XACMLPolicy	69
ArcSec::XACMLRequest	70
ArcSec::XACMLRule	70
ArcSec::XACMLTarget	71
ArcSec::XACMLTargetMatch	71
ArcSec::XACMLTargetMatchGroup	71
ArcSec::XACMLTargetSection	71

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

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

ArcSec::GACLEvaluator	24
ArcSec::GACLPDP	24
ArcSec::GACLPolicy	25
ArcSec::GACLRequest	25
Arc::HTTPResponseHeader	25
Arc::HTTPSClient	25
Arc::HTTPSClientConnector	25
Arc::HTTPSClientConnectorGlobus	26
Arc::HTTPSClientConnectorGSSAPI	27
Arc::HTTPSClientSOAP	28
Arc::LDAPQuery	28
Arc::Lister	29
Arc::MCC_GSI_Client	29
Arc::MCC_GSI_Service	29
Arc::MCC_HTTP (A base class for HTTP client and service MCCs)	30
Arc::MCC_HTTP_Client	30
Arc::MCC_HTTP_Service	31
Arc::MCC_SOAP (A base class for SOAP client and service MCCs)	31
Arc::MCC_SOAP_Client	32
Arc::MCC_SOAP_Service	32
Arc::MCC_TCP (A base class for TCP client and service MCCs)	32
Arc::MCC_TCP_Client	33
Arc::MCC_TCP_Service	33
Arc::MCC_TLS (A base class for TLS client and service MCCs)	34
Arc::MCC_TLS_Client	35
Arc::MCC_TLS_Service	35
Arc::PayloadGSISStream	36
Arc::PayloadHTTP	36
Arc::PayloadTCPSocket	40
Arc::PayloadTLSMCC	41
Arc::PayloadTLSStream	42
ArcSec::PDPSERVICEInvoker (PDPSERVICEInvoker (p. 43) - client which will invoke pdpservice)	43
ArcSec::SAML2SSO_AssertionConsumerSH (Implement the functionality of the Service Provider in SAML2 SSO profile)	43
ArcSec::SAMLTokenSH (Adds WS-Security SAML Token into SOAP Header)	44
ArcSec::SimpleListPDP (Tests X509 subject against list of subjects in file)	44
SRM1Client	44
SRM22Client	50
SRMClient	52
SRMClientRequest	60
SRMFileInfo	62
SRMFileMetaData	62
SRMInfo	62
SRMInvalidRequestException	63
SRMURL	63
ArcSec::UsernameTokenSH (Adds WS-Security Username Token into SOAP Header)	64
ArcSec::X509TokenSH (Adds WS-Security X509 Token into SOAP Header)	64
ArcSec::XACMLAlgFactory (Algorithm factory class for XACML)	65
ArcSec::XACMLApply	65
ArcSec::XACMLAttributeFactory (Attribute factory class for XACML specified attributes) .	65
ArcSec::XACMLAttributeProxy< TheAttribute > (XACML specific AttributeProxy class) .	66
ArcSec::XACMLCondition (XACMLCondition (p. 66) class to parse and operate XACML specific <Condition> node)	66

ArcSec::XACMLEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	67
ArcSec::XACMLEvaluator (Execute the policy evaluation, based on the request and policy) .	67
ArcSec::XACMLFnFactory (Function factory class for XACML specified attributes)	68
ArcSec::XACMLPDP (XACMLPDP (p. 69) - PDP which can handle the XACML specific request and policy schema)	69
ArcSec::XACMLPolicy (XACMLPolicy (p. 69) class to parse and operate XACML specific <Policy> node)	69
ArcSec::XACMLRequest	70
ArcSec::XACMLRule (XACMLRule (p. 70) class to parse XACML specific <Rule> node) .	70
ArcSec::XACMLTarget (XACMLTarget (p. 71) class to parse and operate XACML specific <Target> node)	71
ArcSec::XACMLTargetMatch	71
ArcSec::XACMLTargetMatchGroup	71
ArcSec::XACMLTargetSection	71

Chapter 4

Namespace Documentation

4.1 ArcSec Namespace Reference

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

Data Structures

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

This PDP always return true (allow).

- class **ArcAuthZ**

Tests message against list of PDPs.

- class **ArcAlgFactory**

Algorithm factory class for Arc.

- class **ArcAttributeFactory**

Attribute factory class for Arc specified attributes.

- class **ArcAttributeProxy**

Arc specific AttributeProxy class.

- class **ArcRequestTuple**

RequestTuple, container which includes the.

- class **ArcEvaluationCtx**

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

- class **ArcEvaluator**

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

- class **ArcFnFactory**
Function factory class for Arc specified attributes.
- class **ArcPDP**
ArcPDP (p. 19) - PDP which can handle the Arc specific request and policy schema.
- class **ArcPolicy**
ArcPolicy (p. 20) class to parse and operate Arc specific <Policy> node.
- class **ArcRequest**
- class **ArcRequestItem**
Container, <Subjects, Actions, Objects, Contexts> tuple.
- class **ArcRule**
ArcRule (p. 21) class to parse Arc specific <Rule> node.
- class **DelegationPDP**
- class **DelegationSH**
- class **DenyPDP**
This PDP always returns false (deny).
- class **GACLEvaluator**
- class **GACLPDP**
- class **GACLPolicy**
- class **GACLRequest**
- class **PDPServiceInvoker**
PDPServiceInvoker (p. 43) - client which will invoke pdpservice.
- class **SAML2SSO_AssertionConsumerSH**
Implement the functionality of the Service Provider in SAML2 SSO profile.
- class **SAMLTokenSH**
Adds WS-Security SAML Token into SOAP Header.
- class **SimpleListPDP**
Tests X509 subject against list of subjects in file.
- class **UsernameTokenSH**
Adds WS-Security Username Token into SOAP Header.
- class **X509TokenSH**
Adds WS-Security X509 Token into SOAP Header.
- class **AttributeDesignator**
- class **AttributeSelector**
- class **XACMLAlgFactory**
Algorithm factory class for XACML.
- class **XACMLApply**
- class **XACMLAttributeFactory**

Attribute factory class for XACML specified attributes.

- class **XACMLAttributeProxy**

XACML specific AttributeProxy class.

- class **XACMLCondition**

XACMLCondition (p. 66) class to parse and operate XACML specific <Condition> node.

- class **XACMLEvaluationCtx**

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

- class **XACMLEvaluator**

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

- class **XACMLFnFactory**

Function factory class for XACML specified attributes.

- class **XACMLPDP**

XACMLPDP (p. 69) - PDP which can handle the XACML specific request and policy schema.

- class **XACMLPolicy**

XACMLPolicy (p. 69) class to parse and operate XACML specific <Policy> node.

- class **XACMLRequest**

- class **XACMLRule**

XACMLRule (p. 70) class to parse XACML specific <Rule> node.

- class **XACMLTargetMatch**

- class **XACMLTargetMatchGroup**

- class **XACMLTargetSection**

- class **XACMLTarget**

XACMLTarget (p. 71) class to parse and operate XACML specific <Target> node.

Typedefs

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

4.1.1 Detailed Description

ArcRequest (p. 20), Parsing the specified Arc request format. **XACMLRequest** (p. 70), Parsing the xacml request format.

4.1.2 Typedef Documentation

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

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

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

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

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

Chapter 5

Data Structure Documentation

5.1 ArcSec::AllowPDP Class Reference

This PDP always return true (allow).

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

5.1.1 Detailed Description

This PDP always return true (allow).

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

- AllowPDP.h

5.2 ArcSec::ArcAlgFactory Class Reference

Algorithm factory class for Arc.

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

Public Member Functions

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

5.2.1 Detailed Description

Algorithm factory class for Arc.

5.2.2 Member Function Documentation

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

return a Alg object according to the "CombiningAlg" attribute in the <Policy> node; The **ArcAlgFactory** (p. 15) itself will release the Alg objects

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

- ArcAlgFactory.h

5.3 ArcSec::ArcAttributeFactory Class Reference

Attribute factory class for Arc specified attributes.

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

Public Member Functions

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

5.3.1 Detailed Description

Attribute factory class for Arc specified attributes.

5.3.2 Member Function Documentation

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

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

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

- ArcAttributeFactory.h

5.4 ArcSec::ArcAttributeProxy< TheAttribute > Class Template Reference

Arc specific AttributeProxy class.

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

Public Member Functions

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

5.4.1 Detailed Description

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

Arc specific AttributeProxy class.

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

- ArcAttributeProxy.h

5.5 ArcSec::ArcAuthZ Class Reference

Tests message against list of PDPs.

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

Data Structures

- class PDPDesc

Public Member Functions

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

Protected Member Functions

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

5.5.1 Detailed Description

Tests message against list of PDPs. This class implements SecHandler interface. It's **Handle()** (p. 17) method runs provided Message instance against all PDPs specified in configuration. If any of PDPs returns positive result **Handle()** (p. 17) return true, otherwise false. This class is the main entry for configuring authorization, and could include different PDP configured inside.

5.5.2 Member Function Documentation

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

Get authorization decision

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

Create PDP according to conf info

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

- ArcAuthZ.h

5.6 ArcSec::ArcEvaluationCtx Class Reference

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

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

Public Member Functions

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

5.6.1 Detailed Description

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

5.6.2 Constructor & Destructor Documentation

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

Construct a new EvaluationCtx based on the given request

5.6.3 Member Function Documentation

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

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

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

- ArcEvaluationCtx.h

5.7 ArcSec::ArcEvaluator Class Reference

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

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

Public Member Functions

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

5.7.1 Detailed Description

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

5.7.2 Member Function Documentation

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

Evaluate the request based on the policy information inside PolicyStore

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

- ArcEvaluator.h

5.8 ArcSec::ArcFnFactory Class Reference

Function factory class for Arc specified attributes.

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

Public Member Functions

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

5.8.1 Detailed Description

Function factory class for Arc specified attributes.

5.8.2 Member Function Documentation

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

return a Function object according to the "Function" attribute in the XML node; The **ArcFnFactory** (p. 19) itself will release the Function objects

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

- ArcFnFactory.h

5.9 ArcSec::ArcPDP Class Reference

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

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

5.9.1 Detailed Description

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

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

- ArcPDP.h

5.10 ArcSec::ArcPolicy Class Reference

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

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

Public Member Functions

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

5.10.1 Detailed Description

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

5.10.2 Constructor & Destructor Documentation

5.10.2.1 ArcSec::ArcPolicy::ArcPolicy (void)

Constructor

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

Constructor

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

Constructor

5.10.3 Member Function Documentation

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

Parse XMLNode, and construct the low-level Rule object

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

- ArcPolicy.h

5.11 ArcSec::ArcRequest Class Reference

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

- ArcRequest.h

5.12 ArcSec::ArcRequestItem Class Reference

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

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

5.12.1 Detailed Description

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

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

- ArcRequestItem.h

5.13 ArcSec::ArcRequestTuple Class Reference

RequestTuple, container which includes the.

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

5.13.1 Detailed Description

RequestTuple, container which includes the.

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

- ArcEvaluationCtx.h

5.14 ArcSec::ArcRule Class Reference

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

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

5.14.1 Detailed Description

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

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

- ArcRule.h

5.15 ArcSec::AttributeDesignator Class Reference

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

- AttributeDesignator.h

5.16 ArcSec::AttributeSelector Class Reference

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

- AttributeSelector.h

5.17 Arc::ConfigTlsmcc Class Reference

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

- ConfigTlsmcc.h

5.18 Arc::DataPointArc Class Reference

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

- DataPointArc.h

5.19 Arc::DataPointFile Class Reference

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

- DataPointFile.h

5.20 Arc::DataPointGridFTP Class Reference

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

- DataPointGridFTP.h

5.21 Arc::DataPointHTTP Class Reference

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

- DataPointHTTP.h

5.22 Arc::DataPointLDAP Class Reference

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

- DataPointLDAP.h

5.23 Arc::DataPointLFC Class Reference

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

- DataPointLFC.h

5.24 Arc::DataPointRLS Class Reference

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

- DataPointRLS.h

5.25 Arc::DataPointSRM Class Reference

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

- DataPointSRM.h

5.26 ArcSec::DelegationCollector Class Reference

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

- DelegationCollector.h

5.27 ArcSec::DelegationMultiSecAttr Class Reference

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

- DelegationSecAttr.h

5.28 ArcSec::DelegationPDP Class Reference

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

5.28.1 Detailed Description

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

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

- DelegationPDP.h

5.29 ArcSec::DelegationSecAttr Class Reference

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

- DelegationSecAttr.h

5.30 ArcSec::DelegationSH Class Reference

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

- DelegationSH.h

5.31 ArcSec::DenyPDP Class Reference

This PDP always returns false (deny).

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

5.31.1 Detailed Description

This PDP always returns false (deny).

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

- DenyPDP.h

5.32 ArcSec::GACLEvaluator Class Reference

Public Member Functions

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

5.32.1 Member Function Documentation

5.32.1.1 virtual Response* ArcSec::GACLEvaluator::evaluate (Request * *request*)
[virtual]

Evaluate the request based on the policy information inside PolicyStore

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

- GACLEvaluator.h

5.33 ArcSec::GACLPDP Class Reference

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

- GACLPDP.h

5.34 ArcSec::GACLPolicy Class Reference

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

- GACLPolicy.h

5.35 ArcSec::GACLRequest Class Reference

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

- GACLRequest.h

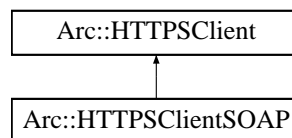
5.36 Arc::HTTPResponseHeader Class Reference

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

- HTTPSCient.h

5.37 Arc::HTTPSCient Class Reference

Inheritance diagram for Arc::HTTPSCient:



Public Member Functions

- int **connect** (void)

5.37.1 Member Function Documentation

5.37.1.1 int Arc::HTTPSCient::connect (void)

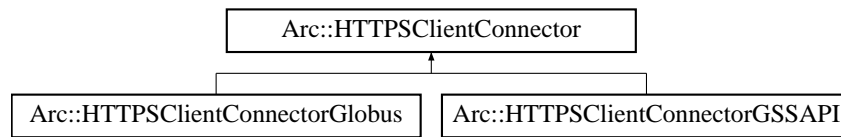
Returns 0 on success, 1 on timeout and -1 on other error

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

- HTTPSCient.h

5.38 Arc::HTTPSCientConnector Class Reference

Inheritance diagram for Arc::HTTPSCientConnector:



Protected Member Functions

- virtual bool **connect** (bool &timedout)
- virtual bool **disconnect** (void)
- virtual bool **clear** (void)
- virtual bool **read** (char *buf=NULL, unsigned int *size=NULL)
- virtual bool **write** (const char *buf=NULL, unsigned int size=0)
- virtual bool **transfer** (bool &read, bool &write, int timeout)
- virtual bool **eofread** (void)
- virtual bool **eofwrite** (void)

Static Protected Attributes

- static SimpleCondition * **connect_lock**
- static Logger **logger**

5.38.1 Member Function Documentation

5.38.1.1 virtual bool Arc::HTTPSClientConnector::transfer (bool & read, bool & write, int timeout) [protected, virtual]

Transfer data set by **read**() (p. 26) and **write**() (p. 26). Reset set buffers if operation complete.

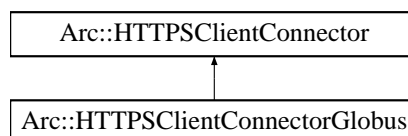
Reimplemented in **Arc::HTTPSClientConnectorGlobus** (p. 27), and **Arc::HTTPSClientConnectorGSSAPI** (p. 28).

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

- HTTPSClient.h

5.39 Arc::HTTPSClientConnectorGlobus Class Reference

Inheritance diagram for Arc::HTTPSClientConnectorGlobus:



Protected Member Functions

- virtual bool **connect** (bool &timedout)
- virtual bool **disconnect** (void)
- virtual bool **read** (char *buf=NULL, unsigned int *size=NULL)
- virtual bool **write** (const char *buf=NULL, unsigned int size=0)
- virtual bool **clear** (void)
- virtual bool **transfer** (bool &read, bool &write, int timeout)
- virtual bool **eofread** (void)
- virtual bool **eofwrite** (void)

5.39.1 Member Function Documentation

5.39.1.1 virtual bool Arc::HTTPSClientConnectorGlobus::transfer (bool & *read*, bool & *write*, int *timeout*) [protected, virtual]

Transfer data set by **read**() (p. 27) and **write**() (p. 27). Reset set buffers if operation complete.

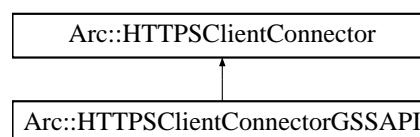
Reimplemented from **Arc::HTTPSClientConnector** (p. 26).

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

- HTTPSClient.h

5.40 Arc::HTTPSClientConnectorGSSAPI Class Reference

Inheritance diagram for Arc::HTTPSClientConnectorGSSAPI:



Protected Member Functions

- virtual bool **connect** (bool &timedout)
- virtual bool **disconnect** (void)
- virtual bool **clear** (void)
- virtual bool **read** (char *buf=NULL, unsigned int *size=NULL)
- virtual bool **write** (const char *buf=NULL, unsigned int size=0)
- virtual bool **transfer** (bool &read, bool &write, int timeout)
- virtual bool **eofread** (void)
- virtual bool **eofwrite** (void)

5.40.1 Member Function Documentation

5.40.1.1 `virtual bool Arc::HTTPSClientConnectorGSSAPI::transfer (bool & read, bool & write, int timeout) [protected, virtual]`

Transfer data set by `read()` (p. 27) and `write()` (p. 27). Reset set buffers if operation complete.

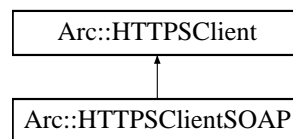
Reimplemented from `Arc::HTTPSClientConnector` (p. 26).

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

- `HTTPSClient.h`

5.41 `Arc::HTTPSClientSOAP` Class Reference

Inheritance diagram for `Arc::HTTPSClientSOAP`:



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

- `HTTPSClient.h`

5.42 `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)

5.42.1 Detailed Description

LDAPQuery (p. 28) class; querying of LDAP servers.

5.42.2 Constructor & Destructor Documentation

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

Constructs a new **LDAPQuery** (p. 28) object and sets connection options. The connection is first established when calling **Query**.

5.42.2.2 Arc::LDAPQuery::~~LDAPQuery ()

Destructor. Will disconnect from the ldapserver if still connected.

5.42.3 Member Function Documentation

5.42.3.1 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.

5.42.3.2 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

5.43 Arc::Lister Class Reference

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

- Lister.h

5.44 Arc::MCC_GSI_Client Class Reference

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

- MCCGSI.h

5.45 Arc::MCC_GSI_Service Class Reference

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

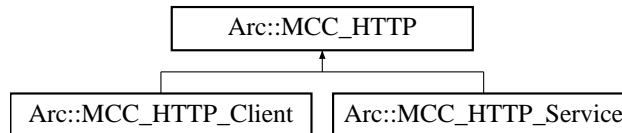
- MCCGSI.h

5.46 Arc::MCC_HTTP Class Reference

A base class for HTTP client and service MCCs.

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

Inheritance diagram for Arc::MCC_HTTP:



5.46.1 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.

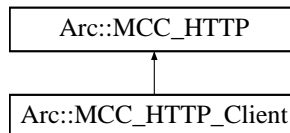
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.47 Arc::MCC_HTTP_Client Class Reference

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

Inheritance diagram for Arc::MCC_HTTP_Client:



5.47.1 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.

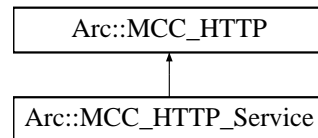
The documentation for this class was generated from the following file:

- MCCHTTP.h

5.48 Arc::MCC_HTTP_Service Class Reference

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

Inheritance diagram for Arc::MCC_HTTP_Service:



5.48.1 Detailed Description

This class implements MCC to processes HTTP request. On input payload with PayloadStreamInterface is expected. HTTP message is read from stream and its body is converted into PayloadRaw and passed to next MCC. Returned payload of PayloadRawInterface type is treated as body part of returning **PayloadHTTP** (p.36). Generated HTTP response is sent through stream passed in input payload. During processing of request/input message following attributes are generated: HTTP:METHORD - 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:

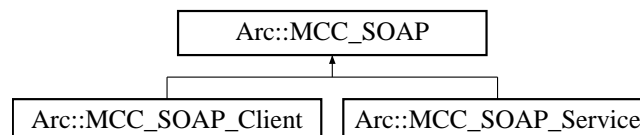
- MCCHTTP.h

5.49 Arc::MCC_SOAP Class Reference

A base class for SOAP client and service MCCs.

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

Inheritance diagram for Arc::MCC_SOAP:



5.49.1 Detailed Description

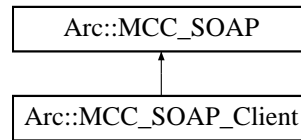
A base class for SOAP client and service MCCs. This is a base class for SOAP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

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

- MCCSOAP.h

5.50 Arc::MCC_SOAP_Client Class Reference

Inheritance diagram for Arc::MCC_SOAP_Client:



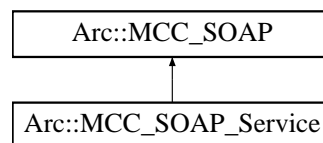
The documentation for this class was generated from the following file:

- MCCSOAP.h

5.51 Arc::MCC_SOAP_Service Class Reference

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

Inheritance diagram for Arc::MCC_SOAP_Service:



5.51.1 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:

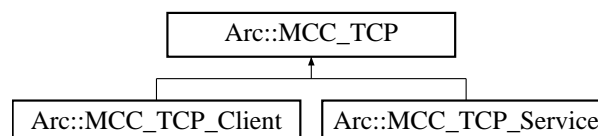
- MCCSOAP.h

5.52 Arc::MCC_TCP Class Reference

A base class for TCP client and service MCCs.

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

Inheritance diagram for Arc::MCC_TCP:



5.52.1 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.

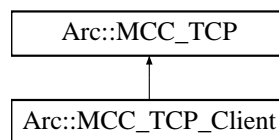
The documentation for this class was generated from the following file:

- MCCTCP.h

5.53 Arc::MCC_TCP_Client Class Reference

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

Inheritance diagram for Arc::MCC_TCP_Client:



5.53.1 Detailed Description

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

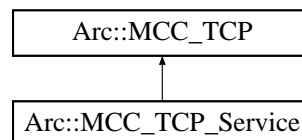
The documentation for this class was generated from the following file:

- MCCTCP.h

5.54 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)

5.54.1 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`

5.54.2 Constructor & Destructor Documentation

5.54.2.1 `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:

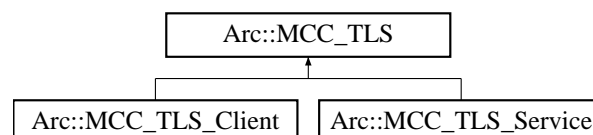
- `MCCTCP.h`

5.55 `Arc::MCC_TLS` Class Reference

A base class for TLS client and service MCCs.

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

Inheritance diagram for `Arc::MCC_TLS`:



5.55.1 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.

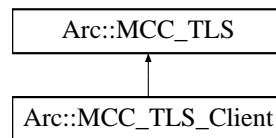
The documentation for this class was generated from the following file:

- `MCCTLS.h`

5.56 Arc::MCC_TLS_Client Class Reference

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

Inheritance diagram for Arc::MCC_TLS_Client:



5.56.1 Detailed Description

This class is MCC implementing TLS client.

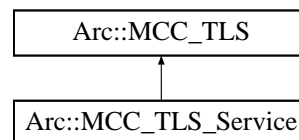
The documentation for this class was generated from the following file:

- MCCTLS.h

5.57 Arc::MCC_TLS_Service Class Reference

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

Inheritance diagram for Arc::MCC_TLS_Service:



5.57.1 Detailed Description

This MCC implements TLS server side functionality. Upon creation this object creates SSL_CTX object and configures SSL_CTX object with some environment information about credential. Because we cannot know the "socket" when the creation of MCC_TLS_Service/MCC_TLS_Client object (not like **MCC_TCP_Client** (p. 33), which can creat socket in the constructor method by using information in configuration file), we can only creat "ssl" object which is binded to specified "socket", when **MCC_HTTP_Client** (p. 30) calls the process() method of **MCC_TLS_Client** (p. 35) object, or **MCC_TCP_Service** (p. 33) calls the process() method of **MCC_TLS_Service** (p. 35) object. The "ssl" object is embeded in a payload called PayloadTLSSocket.

The process() method of **MCC_TLS_Service** (p. 35) 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 remoote peer.

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

- MCCTLS.h

5.58 Arc::PayloadGSISStream Class Reference

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

- PayloadGSISStream.h

5.59 Arc::PayloadHTTP Class Reference

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

Public Member Functions

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

Protected Member Functions

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

Protected Attributes

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

5.59.1 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.

5.59.2 Constructor & Destructor Documentation

5.59.2.1 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.

5.59.2.2 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.

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

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

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

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

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

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

5.59.3 Member Function Documentation

5.59.3.1 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.

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

Adds HTTP header attribute 'name' = 'value'

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

Returns all HTTP header attributes.

5.59.3.4 `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.

5.59.3.5 `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.

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

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

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

Read HTTP header and fill internal variables

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

Read up to 'size' bytes from stream_

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

Read from stream till

5.59.4 Field Documentation

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

true if connection should not be closed after response

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

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

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

Content-length of HTTP message

5.59.4.4 int Arc::PayloadHTTP::code_ [protected]

HTTP method being used or requested

5.59.4.5 bool Arc::PayloadHTTP::keep_alive_ [protected]

true if content is chunked

5.59.4.6 int64_t Arc::PayloadHTTP::length_ [protected]

HTTP reason being sent or supplied

5.59.4.7 std::string Arc::PayloadHTTP::method_ [protected]

minor number of HTTP version - must be 0 or 1

5.59.4.8 PayloadRawInterface* Arc::PayloadHTTP::rbody_ [protected]

if true stream_ is owned by this

5.59.4.9 std::string Arc::PayloadHTTP::reason_ [protected]

HTTP code being sent or supplied

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

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

5.59.4.11 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_.

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

stream used to communicate to outside

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

if true body_ is owned by this

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

URI being contacted

5.59.4.15 `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

5.60 `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)

5.60.1 Detailed Description

This class extends `PayloadStream` with TCP socket specific features

5.60.2 Constructor & Destructor Documentation

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

Constructor - connects to TCP server at specified hostname:port

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

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

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

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

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

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

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

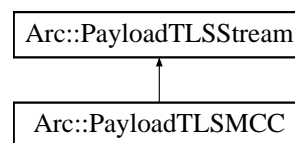
Copy constructor - inherits handle of copied object. Handle is NOT closed in destructor.

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

- PayloadTCPSocket.h

5.61 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)

5.61.1 Constructor & Destructor Documentation

5.61.1.1 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.

5.61.1.2 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.

5.61.1.3 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.

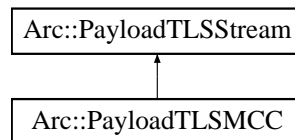
The documentation for this class was generated from the following file:

- PayloadTLMCC.h

5.62 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_

5.62.1 Detailed Description

Implementation of PayloadStreamInterface for SSL handle.

5.62.2 Constructor & Destructor Documentation

5.62.2.1 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.

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

Destructor.

5.62.3 Member Function Documentation

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

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

5.62.3.2 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.

5.62.3.3 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.

5.62.4 Field Documentation

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

Timeout for read/write operations

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

- PayloadTLSStream.h

5.63 ArcSec::PDPSERVICEInvoker Class Reference

PDPSERVICEInvoker (p. 43) - client which will invoke pdpservice.

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

5.63.1 Detailed Description

PDPSERVICEInvoker (p. 43) - client which will invoke pdpservice.

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

- PDPSERVICEInvoker.h

5.64 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference

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

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

5.64.1 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

5.65 ArcSec::SAMLTokenSH Class Reference

Adds WS-Security SAML Token into SOAP Header.

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

5.65.1 Detailed Description

Adds WS-Security SAML Token into SOAP Header.

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

- SAMLTokenSH.h

5.66 ArcSec::SimpleListPDP Class Reference

Tests X509 subject against list of subjects in file.

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

5.66.1 Detailed Description

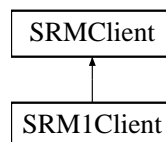
Tests X509 subject against list of subjects in file. This class implements PDP interface. It's isPermitted() method compares X509 subject of requestor obtained from TLS layer (TLS:PEERDN) to list of subjects (one per line) in external file. Location of file is defined by 'location' attribute of PDP configuration.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

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

- SimpleListPDP.h

5.67 SRM1Client Class Reference

Inheritance diagram for SRM1Client:



Public Member Functions

- SRMReturnCode **ping** (std::string &, bool=true)
- SRMReturnCode **getSpaceTokens** (std::list< std::string > &, std::string="")
- SRMReturnCode **getRequestTokens** (std::list< std::string > &, std::string="")
- SRMReturnCode **requestBringOnline** (SRMClientRequest &)
- SRMReturnCode **requestBringOnlineStatus** (SRMClientRequest &)

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

5.67.1 Member Function Documentation

5.67.1.1 SRMReturnCode SRM1Client::abort (SRMClientRequest & *req*) [virtual]

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

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

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

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

Copy a file between two SRM storages.

Parameters

req The request object

source The source URL

Returns

SRMReturnCode specifying outcome of operation

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

5.67.1.3 SRMReturnCode SRM1Client::getRequestTokens (std::list< std::string > & *tokens*, std::string *description* = "") [inline, virtual]

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

Parameters

tokens The list filled by the service

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

Returns

SRMReturnCode specifying outcome of operation

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

5.67.1.4 SRMReturnCode SRM1Client::getSpaceTokens (std::list< std::string > & *tokens*, std::string *description* = "") [inline, virtual]

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

Parameters

tokens The list filled by the service

description The space token description

Returns

SRMReturnCode specifying outcome of operation

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

5.67.1.5 SRMReturnCode SRM1Client::getTURLs (SRMClientRequest & *req*, std::list< std::string > & *urls*) [virtual]

If the user wishes to copy a file from somewhere, **getTURLs()** (p. 46) is called to retrieve the transport URL to copy the file from.

Parameters

req The request object

urls A list of TURLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

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

5.67.1.6 SRMReturnCode SRM1Client::info (SRMClientRequest & *req*, std::list< struct SRMFileMetaData > & *metadata*, const int *recursive* = 0, bool *report_error* = true) [virtual]

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

Parameters

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

Returns

SRMReturnCode specifying outcome of operation

See also

SRMFileMetaData (p. 62)

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

5.67.1.7 SRMReturnCode SRM1Client::mkDir (SRMClientRequest & req) [inline, virtual]

Make required directories for the SURL in the request

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

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

5.67.1.8 SRMReturnCode SRM1Client::ping (std::string & version, bool report_error = true) [inline, virtual]

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

Parameters

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

Returns

SRMReturnCode specifying outcome of operation

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

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

If the user wishes to copy a file to somewhere, **putURLs()** (p. 48) is called to retrieve the transport URL to copy the file to.

Parameters

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

Returns

SRMReturnCode specifying outcome of operation

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

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

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

Parameters

- req* The request object

Returns

SRMReturnCode specifying outcome of operation

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

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

Should be called after a successful copy from SRM storage.

Parameters

- req* The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **SRMClient** (p. 58).

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

Should be called after a successful copy to SRM storage.

Parameters

- req* The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **SRMClient** (p. 58).

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

Delete a file physically from storage and the SRM namespace.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **SRMClient** (p. 58).

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

Submit a request to bring online files. This operation is asynchronous and the status of the request can be checked by calling **requestBringOnlineStatus()** (p. 49) with the request token in *req* which is assigned by this method.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implements **SRMClient** (p. 59).

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

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

Parameters

req The request object to query the status of

Returns

SRMReturnCode specifying outcome of operation

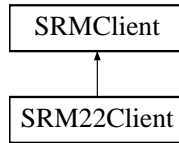
Implements **SRMClient** (p. 59).

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

- SRM1Client.h

5.68 SRM22Client Class Reference

Inheritance diagram for SRM22Client:



Public Member Functions

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

5.68.1 Member Function Documentation

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

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

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

5.68.1.2 SRMReturnCode 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 **SRMClient** (p. 54).

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

Use srmGetRequestTokens to return a list of spaces available

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

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

Use srmGetSpaceTokens to return a list of spaces available

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

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

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

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

5.68.1.6 SRMReturnCode 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 **SRMClient** (p. 56).

5.68.1.7 SRMReturnCode SRM22Client::mkDir (SRMClientRequest & *req*) [virtual]

Call srmMkDir

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

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

Get the server version from srmPing

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

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

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

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

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

Not used in this version of SRM

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

5.68.1.11 SRMReturnCode SRM22Client::releaseGet (SRMClientRequest & *req*) [virtual]

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

Implements **SRMClient** (p. 58).

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

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

Implements **SRMClient** (p. 58).

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

Delete by srmRm or srmRmDir

Implements **SRMClient** (p. 58).

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

Call srmBringOnline with the URLs specified in req.

Implements **SRMClient** (p. 59).

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

Call srmStatusOfBringOnlineRequest and update req with any changes.

Implements **SRMClient** (p. 59).

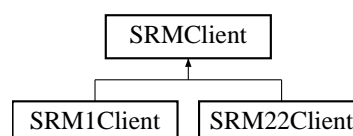
The documentation for this class was generated from the following file:

- SRM22Client.h

5.69 SRMClient Class Reference

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

Inheritance diagram for SRMClient:



Public Member Functions

- SRMReturnCode **connect** (void)
- bool **disconnect** (void)
- virtual ~SRMClient ()
- void **Timeout** (int t)
- std::string **getVersion** () const
- virtual SRMReturnCode **ping** (std::string &version, bool report_error=true)=0
- virtual SRMReturnCode **getSpaceTokens** (std::list< std::string > &tokens, std::string description="")=0
- virtual SRMReturnCode **getRequestTokens** (std::list< std::string > &tokens, std::string description="")=0
- virtual SRMReturnCode **getTURLs** (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode **requestBringOnline** (SRMClientRequest &req)=0
- virtual SRMReturnCode **requestBringOnlineStatus** (SRMClientRequest &req)=0
- virtual SRMReturnCode **putTURLs** (SRMClientRequest &req, std::list< std::string > &urls, unsigned long long size=0)=0
- virtual SRMReturnCode **releaseGet** (SRMClientRequest &req)=0
- virtual SRMReturnCode **releasePut** (SRMClientRequest &req)=0
- virtual SRMReturnCode **release** (SRMClientRequest &req)=0
- virtual SRMReturnCode **abort** (SRMClientRequest &req)=0
- virtual SRMReturnCode **info** (SRMClientRequest &req, std::list< struct SRMFileMetaData > &metadata, const int recursive=0, bool report_error=true)=0
- virtual SRMReturnCode **remove** (SRMClientRequest &req)=0
- virtual SRMReturnCode **copy** (SRMClientRequest &req, const std::string &source)=0
- virtual SRMReturnCode **mkDir** (SRMClientRequest &req)=0

Static Public Member Functions

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

Protected Attributes

- std::string **service_endpoint**
- Arc::HTTPSCClientSOAP * **csoap**
- SRMImplementation **implementation**
- std::string **version**

Static Protected Attributes

- static time_t **request_timeout**
- static Arc::Logger **logger**

5.69.1 Detailed Description

A client interface to the SRM protocol. Instances of SRM clients are created by calling the **getInstance()** (p. 55) 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.

5.69.2 Constructor & Destructor Documentation

5.69.2.1 `virtual SRMClient::~~SRMClient () [inline, virtual]`

empty destructor

5.69.3 Member Function Documentation

5.69.3.1 `virtual SRMReturnCode SRMClient::abort (SRMClientRequest & req) [pure virtual]`

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

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 45), and **SRM22Client** (p. 50).

5.69.3.2 `SRMReturnCode SRMClient::connect (void)`

Establish a connection to the service

Returns

SRMReturnCode specifying outcome of operation

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

Copy a file between two SRM storages.

Parameters

req The request object

source The source SURL

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 45), and **SRM22Client** (p. 50).

5.69.3.4 `bool SRMClient::disconnect (void) [inline]`

Disconnect from the service and destroy the connection

References csoap.

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

Returns an **SRMClient** (p. 52) instance with the required protocol version. This must be used to create **SRMClient** (p. 52) instances. Specifying a version explicitly forces creation of a client with that version.

Parameters

usercfg The user configuration.

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

timedout Whether the connection timed out

timeout Connection timeout. is returned.

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

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

Parameters

tokens The list filled by the service

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

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 45), and **SRM22Client** (p. 50).

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

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

Parameters

tokens The list filled by the service

description The space token description

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 46), and **SRM22Client** (p. 51).

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

If the user wishes to copy a file from somewhere, `getTURLs()` (p. 56) is called to retrieve the transport URL to copy the file from.

Parameters

req The request object

urls A list of TURLs filled by the method

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 46), and **SRM22Client** (p. 51).

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

Returns the version of the SRM protocol used by this instance

References version.

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

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

Parameters

req The request object

metadata A list of structs filled with file information

recursive The level of recursion into sub directories

report_error Determines if errors should be reported

Returns

SRMReturnCode specifying outcome of operation

See also

SRMFileMetaData (p. 62)

Implemented in **SRM1Client** (p. 46), and **SRM22Client** (p. 51).

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

Make required directories for the SURL in the request

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 47), and **SRM22Client** (p. 51).

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

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

Parameters

version The version returned by the server

report_error Whether an error should be reported

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 47), and **SRM22Client** (p. 51).

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

If the user wishes to copy a file to somewhere, **putURLs()** (p. 57) is called to retrieve the transport URL to copy the file to.

Parameters

req The request object

urls A list of TURLs filled by the method

size The size of the file

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 48), and **SRM22Client** (p. 51).

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

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

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 48), and **SRM22Client** (p. 51).

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

Should be called after a successful copy from SRM storage.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 48), and **SRM22Client** (p. 52).

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

Should be called after a successful copy to SRM storage.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 48), and **SRM22Client** (p. 52).

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

Delete a file physically from storage and the SRM namespace.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 49), and **SRM22Client** (p. 52).

5.69.3.18 virtual SRMReturnCode SRMClient::requestBringOnline (SRMClientRequest & req) [pure virtual]

Submit a request to bring online files. This operation is asynchronous and the status of the request can be checked by calling **requestBringOnlineStatus()** (p. 59) with the request token in req which is assigned by this method.

Parameters

req The request object

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 49), and **SRM22Client** (p. 52).

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

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

Parameters

req The request object to query the status of

Returns

SRMReturnCode specifying outcome of operation

Implemented in **SRM1Client** (p. 49), and **SRM22Client** (p. 52).

5.69.3.20 void SRMClient::Timeout (int t) [inline]

set the request timeout

References request_timeout.

5.69.4 Field Documentation**5.69.4.1 Arc::HTTPSClientSOAP* SRMClient::csoap [protected]**

SOAP client object

Referenced by disconnect().

5.69.4.2 SRMImplementation SRMClient::implementation [protected]

The implementation of the server

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

Logger

5.69.4.4 `time_t SRMClient::request_timeout` [static, protected]

Timeout for requests to the SRM service

Referenced by `Timeout()`.

5.69.4.5 `std::string 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.

5.69.4.6 `std::string 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`

5.70 SRMClientRequest Class Reference

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

Public Member Functions

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

5.70.1 Detailed Description

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

5.70.2 Constructor & Destructor Documentation

5.70.2.1 SRMClientRequest::SRMClientRequest (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/pnfs/ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/dummyfile3

5.70.2.2 SRMClientRequest::SRMClientRequest (std::string *url* = "", 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/pnfs/ndgf.org/data/atlas/disk/user/user.mlassnig.dataset.1/dummyfile3

5.70.3 Member Function Documentation

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

set and get file id list

5.70.3.2 void SRMClientRequest::finished_success () [inline]

set and get status of request

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

set and get long list flag

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

set and get request id

5.70.3.5 void SRMClientRequest::request_token (char * *token*) [inline]

set and get request token

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

set and get space token

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

set and get surl failures

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

set and get surl statuses

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

get URLs

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

set and get waiting time

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

- SRMClient.h

5.71 SRMFileInfo Class Reference

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

5.71.1 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

5.72 SRMFileMetaData Struct Reference

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

5.72.1 Detailed Description

File metadata

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

- SRMClient.h

5.73 SRMInfo Class Reference

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

5.73.1 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

5.74 SRMInvalidRequestException Class Reference

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

- SRMClient.h

5.75 SRMURL Class Reference

Public Member Functions

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

5.75.1 Constructor & Destructor Documentation

5.75.1.1 SRMURL::SRMURL (std::string url)

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

5.75.2 Member Function Documentation

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

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

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

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

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

eg `/srm/managerv2`

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

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

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

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

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

Was the port number given in the constructor?

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

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

5.75.2.8 `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

5.76 ArcSec::UsernameTokenSH Class Reference

Adds WS-Security Username Token into SOAP Header.

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

5.76.1 Detailed Description

Adds WS-Security Username Token into SOAP Header.

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

- UsernameTokenSH.h

5.77 ArcSec::X509TokenSH Class Reference

Adds WS-Security X509 Token into SOAP Header.

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

5.77.1 Detailed Description

Adds WS-Security X509 Token into SOAP Header.

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

- X509TokenSH.h

5.78 ArcSec::XACMLAlgFactory Class Reference

Algorithm factory class for XACML.

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

Public Member Functions

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

5.78.1 Detailed Description

Algorithm factory class for XACML.

5.78.2 Member Function Documentation

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

return a Alg object according to the "CombiningAlg" attribute in the <Policy> node; The **XACMLAlgFactory** (p. 65) itself will release the Alg objects

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

- XACMLAlgFactory.h

5.79 ArcSec::XACMLApply Class Reference

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

- XACMLApply.h

5.80 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)

5.80.1 Detailed Description

Attribute factory class for XACML specified attributes.

5.80.2 Member Function Documentation

5.80.2.1 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

5.81 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference

XACML specific AttributeProxy class.

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

Public Member Functions

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

5.81.1 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

5.82 ArcSec::XACMLCondition Class Reference

XACMLCondition (p. 66) class to parse and operate XACML specific <Condition> node.

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


Public Member Functions

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

5.82.1 Detailed Description

XACMLCondition (p. 66) class to parse and operate XACML specific <Condition> node.

5.82.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLCondition.h

5.83 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)

5.83.1 Detailed Description

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

5.83.2 Constructor & Destructor Documentation

5.83.2.1 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

5.84 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)

5.84.1 Detailed Description

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

5.84.2 Member Function Documentation

5.84.2.1 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

5.85 ArcSec::XACMLFnFactory Class Reference

Function factory class for XACML specified attributes.

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

Public Member Functions

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

5.85.1 Detailed Description

Function factory class for XACML specified attributes.

5.85.2 Member Function Documentation

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

return a Function object according to the "Function" attribute in the XML node; The **XACMLFnFactory** (p. 68) itself will release the Function objects

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

- XACMLFnFactory.h

5.86 ArcSec::XACMLPDP Class Reference

XACMLPDP (p. 69) - PDP which can handle the XACML specific request and policy schema.

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

5.86.1 Detailed Description

XACMLPDP (p. 69) - PDP which can handle the XACML specific request and policy schema.

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

- XACMLPDP.h

5.87 ArcSec::XACMLPolicy Class Reference

XACMLPolicy (p. 69) 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** ()

5.87.1 Detailed Description

XACMLPolicy (p. 69) class to parse and operate XACML specific <Policy> node.

5.87.2 Constructor & Destructor Documentation

5.87.2.1 ArcSec::XACMLPolicy::XACMLPolicy (void)

Constructor

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

Constructor

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

Constructor -

5.87.3 Member Function Documentation

5.87.3.1 `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

5.88 ArcSec::XACMLRequest Class Reference

Public Member Functions

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

5.88.1 Member Function Documentation

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

Get the name of corresponding evaluator

5.88.1.2 `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

5.89 ArcSec::XACMLRule Class Reference

XACMLRule (p. 70) class to parse XACML specific <Rule> node.

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

5.89.1 Detailed Description

XACMLRule (p. 70) class to parse XACML specific <Rule> node.

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

- XACMLRule.h

5.90 ArcSec::XACMLTarget Class Reference

XACMLTarget (p. 71) class to parse and operate XACML specific <Target> node.

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

Public Member Functions

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

5.90.1 Detailed Description

XACMLTarget (p. 71) class to parse and operate XACML specific <Target> node.

5.90.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLTarget.h

5.91 ArcSec::XACMLTargetMatch Class Reference

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

- XACMLTarget.h

5.92 ArcSec::XACMLTargetMatchGroup Class Reference

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

- XACMLTarget.h

5.93 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, 42
- ~SRMClient
 - SRMClient, 54
- abort
 - SRM1Client, 45
 - SRM22Client, 50
 - SRMClient, 54
- AndList
 - ArcSec, 14
- Arc::ConfigTLSMCC, 22
- Arc::DataPointARC, 22
- Arc::DataPointFile, 22
- Arc::DataPointGridFTP, 22
- Arc::DataPointHTTP, 22
- Arc::DataPointLDAP, 22
- Arc::DataPointLFC, 23
- Arc::DataPointRLS, 23
- Arc::DataPointSRM, 23
- Arc::HTTPResponseHeader, 25
- Arc::HTTPSCient, 25
 - connect, 25
- Arc::HTTPSCientConnector, 25
 - transfer, 26
- Arc::HTTPSCientConnectorGlobus, 26
 - transfer, 27
- Arc::HTTPSCientConnectorGSSAPI, 27
 - transfer, 28
- Arc::HTTPSCientSOAP, 28
- Arc::LDAPQuery, 28
 - ~LDAPQuery, 29
 - LDAPQuery, 29
 - Query, 29
 - Result, 29
- Arc::Lister, 29
- Arc::MCC_GSI_Client, 29
- Arc::MCC_GSI_Service, 29
- Arc::MCC_HTTP, 30
- Arc::MCC_HTTP_Client, 30
- Arc::MCC_HTTP_Service, 31
- Arc::MCC_SOAP, 31
- Arc::MCC_SOAP_Client, 32
- Arc::MCC_SOAP_Service, 32
- Arc::MCC_TCP, 32
- Arc::MCC_TCP_Client, 33
- Arc::MCC_TCP_Service, 33
 - MCC_TCP_Service, 34
- Arc::MCC_TLS, 34
- Arc::MCC_TLS_Client, 35
- Arc::MCC_TLS_Service, 35
- Arc::PayloadGSISStream, 36
- Arc::PayloadHTTP, 36
 - Attribute, 37
 - Attributes, 37
 - attributes_, 38
 - Body, 38
 - body_own_, 38
 - chunked_, 38
 - code_, 38
 - Flush, 38
 - get_body, 38
 - keep_alive_, 39
 - length_, 39
 - method_, 39
 - parse_header, 38
 - PayloadHTTP, 37
 - rbody_, 39
 - read, 38
 - readline, 38
 - reason_, 39
 - sbody_, 39
 - stream_, 39
 - stream_own_, 39
 - uri_, 39
 - version_major_, 39
 - version_minor_, 39
- Arc::PayloadTCPSTicket, 40
 - PayloadTCPSTicket, 40
- Arc::PayloadTLSMCC, 41
 - PayloadTLSMCC, 41
- Arc::PayloadTLSStream, 42
 - ~PayloadTLSStream, 42
 - GetCert, 42
 - GetPeerCert, 42
 - PayloadTLSStream, 42
 - ssl_, 43
 - STACK_OF, 43

- ArcEvaluationCtx
 - ArcSec::ArcEvaluationCtx, 18
- ArcPolicy
 - ArcSec::ArcPolicy, 20
- ArcSec, 11
 - AndList, 14
 - Match, 14
- ArcSec::AllowPDP, 15
- ArcSec::ArcAlgFactory, 15
 - createAlg, 16
- ArcSec::ArcAttributeFactory, 16
 - createValue, 16
- ArcSec::ArcAttributeProxy, 16
- ArcSec::ArcAuthZ, 17
 - Handle, 17
 - MakePDPs, 17
- ArcSec::ArcEvaluationCtx, 18
 - ArcEvaluationCtx, 18
 - split, 18
- ArcSec::ArcEvaluator, 18
 - evaluate, 19
- ArcSec::ArcFnFactory, 19
 - createFn, 19
- ArcSec::ArcPDP, 19
- ArcSec::ArcPolicy, 20
 - ArcPolicy, 20
 - make_policy, 20
- ArcSec::ArcRequest, 20
- ArcSec::ArcRequestItem, 21
- ArcSec::ArcRequestTuple, 21
- ArcSec::ArcRule, 21
- ArcSec::AttributeDesignator, 21
- ArcSec::AttributeSelector, 22
- ArcSec::DelegationCollector, 23
- ArcSec::DelegationMultiSecAttr, 23
- ArcSec::DelegationPDP, 23
- ArcSec::DelegationSecAttr, 24
- ArcSec::DelegationSH, 24
- ArcSec::DenyPDP, 24
- ArcSec::GACLEvaluator, 24
 - evaluate, 24
- ArcSec::GACLPDP, 24
- ArcSec::GACLPolicy, 25
- ArcSec::GACLRequest, 25
- ArcSec::PDPSERVICEInvoker, 43
- ArcSec::SAML2SSO_AssertionConsumerSH, 43
- ArcSec::SAMLTokenSH, 44
- ArcSec::SimpleListPDP, 44
- ArcSec::UsernameTokenSH, 64
- ArcSec::X509TokenSH, 64
- ArcSec::XACMLAlgFactory, 65
 - createAlg, 65
- ArcSec::XACMLApply, 65
- ArcSec::XACMLAttributeFactory, 65
 - createValue, 66
- ArcSec::XACMLAttributeProxy, 66
- ArcSec::XACMLCondition, 66
 - XACMLCondition, 67
- ArcSec::XACMLEvaluationCtx, 67
 - XACMLEvaluationCtx, 67
- ArcSec::XACMLEvaluator, 67
 - evaluate, 68
- ArcSec::XACMLFnFactory, 68
 - createFn, 68
- ArcSec::XACMLPDP, 69
- ArcSec::XACMLPolicy, 69
 - make_policy, 70
 - XACMLPolicy, 69
- ArcSec::XACMLRequest, 70
 - getEvalName, 70
 - getName, 70
- ArcSec::XACMLRule, 70
- ArcSec::XACMLTarget, 71
 - XACMLTarget, 71
- ArcSec::XACMLTargetMatch, 71
- ArcSec::XACMLTargetMatchGroup, 71
- ArcSec::XACMLTargetSection, 71
- Attribute
 - Arc::PayloadHTTP, 37
- Attributes
 - Arc::PayloadHTTP, 37
- attributes_
 - Arc::PayloadHTTP, 38
- BaseURL
 - SRMURL, 63
- Body
 - Arc::PayloadHTTP, 38
- body_own_
 - Arc::PayloadHTTP, 38
- chunked_
 - Arc::PayloadHTTP, 38
- code_
 - Arc::PayloadHTTP, 38
- connect
 - Arc::HTTPSCClient, 25
 - SRMClient, 54
- ContactURL
 - SRMURL, 63
- copy
 - SRM1Client, 45
 - SRM22Client, 50
 - SRMClient, 54
- createAlg
 - ArcSec::ArcAlgFactory, 16
 - ArcSec::XACMLAlgFactory, 65
- createFn

- ArcSec::ArcFnFactory, 19
- ArcSec::XACMLFnFactory, 68
- createValue
 - ArcSec::ArcAttributeFactory, 16
 - ArcSec::XACMLAttributeFactory, 66
- csoap
 - SRMClient, 59
- disconnect
 - SRMClient, 54
- Endpoint
 - SRMURL, 63
- evaluate
 - ArcSec::ArcEvaluator, 19
 - ArcSec::GACLEvaluator, 24
 - ArcSec::XACMLEvaluator, 68
- file_ids
 - SRMClientRequest, 61
- FileName
 - SRMURL, 63
- finished_success
 - SRMClientRequest, 61
- Flush
 - Arc::PayloadHTTP, 38
- FullURL
 - SRMURL, 64
- get_body
 - Arc::PayloadHTTP, 38
- GetCert
 - Arc::PayloadTLSStream, 42
- getEvalName
 - ArcSec::XACMLRequest, 70
- getInstance
 - SRMClient, 54
- getName
 - ArcSec::XACMLRequest, 70
- GetPeerCert
 - Arc::PayloadTLSStream, 42
- getRequestTokens
 - SRM1Client, 45
 - SRM22Client, 50
 - SRMClient, 55
- getSpaceTokens
 - SRM1Client, 46
 - SRM22Client, 50
 - SRMClient, 55
- getURLs
 - SRM1Client, 46
 - SRM22Client, 51
 - SRMClient, 55
- getVersion
 - SRMClient, 56
- Handle
 - ArcSec::ArcAuthZ, 17
- implementation
 - SRMClient, 59
- info
 - SRM1Client, 46
 - SRM22Client, 51
 - SRMClient, 56
- keep_alive_
 - Arc::PayloadHTTP, 39
- LDAPQuery
 - Arc::LDAPQuery, 29
- length_
 - Arc::PayloadHTTP, 39
- logger
 - SRMClient, 59
- long_list
 - SRMClientRequest, 61
- make_policy
 - ArcSec::ArcPolicy, 20
 - ArcSec::XACMLPolicy, 70
- MakePDPs
 - ArcSec::ArcAuthZ, 17
- Match
 - ArcSec, 14
- MCC_TCP_Service
 - Arc::MCC_TCP_Service, 34
- method_
 - Arc::PayloadHTTP, 39
- mkDir
 - SRM1Client, 47
 - SRM22Client, 51
 - SRMClient, 56
- parse_header
 - Arc::PayloadHTTP, 38
- PayloadHTTP
 - Arc::PayloadHTTP, 37
- PayloadTCPSocket
 - Arc::PayloadTCPSocket, 40
- PayloadTLMCC
 - Arc::PayloadTLMCC, 41
- PayloadTLSStream
 - Arc::PayloadTLSStream, 42
- ping
 - SRM1Client, 47
 - SRM22Client, 51
 - SRMClient, 57
- PortDefined

- SRMURL, 64
- putURLs
 - SRM1Client, 47
 - SRM22Client, 51
 - SRMClient, 57
- Query
 - Arc::LDAPQuery, 29
- rbody_
 - Arc::PayloadHTTP, 39
- read
 - Arc::PayloadHTTP, 38
- readline
 - Arc::PayloadHTTP, 38
- reason_
 - Arc::PayloadHTTP, 39
- release
 - SRM1Client, 48
 - SRM22Client, 51
 - SRMClient, 57
- releaseGet
 - SRM1Client, 48
 - SRM22Client, 51
 - SRMClient, 58
- releasePut
 - SRM1Client, 48
 - SRM22Client, 52
 - SRMClient, 58
- remove
 - SRM1Client, 49
 - SRM22Client, 52
 - SRMClient, 58
- request_id
 - SRMClientRequest, 61
- request_timeout
 - SRMClient, 59
- request_token
 - SRMClientRequest, 61
- requestBringOnline
 - SRM1Client, 49
 - SRM22Client, 52
 - SRMClient, 58
- requestBringOnlineStatus
 - SRM1Client, 49
 - SRM22Client, 52
 - SRMClient, 59
- Result
 - Arc::LDAPQuery, 29
- sbody_
 - Arc::PayloadHTTP, 39
- service_endpoint
 - SRMClient, 60
- SetSRMVersion
 - SRMURL, 64
- ShortURL
 - SRMURL, 64
- space_token
 - SRMClientRequest, 61
- split
 - ArcSec::ArcEvaluationCtx, 18
- SRM1Client, 44
 - abort, 45
 - copy, 45
 - getRequestTokens, 45
 - getSpaceTokens, 46
 - getURLs, 46
 - info, 46
 - mkdir, 47
 - ping, 47
 - putURLs, 47
 - release, 48
 - releaseGet, 48
 - releasePut, 48
 - remove, 49
 - requestBringOnline, 49
 - requestBringOnlineStatus, 49
- SRM22Client, 50
 - abort, 50
 - copy, 50
 - getRequestTokens, 50
 - getSpaceTokens, 50
 - getURLs, 51
 - info, 51
 - mkdir, 51
 - ping, 51
 - putURLs, 51
 - release, 51
 - releaseGet, 51
 - releasePut, 52
 - remove, 52
 - requestBringOnline, 52
 - requestBringOnlineStatus, 52
- SRMClient, 52
 - ~SRMClient, 54
 - abort, 54
 - connect, 54
 - copy, 54
 - csoap, 59
 - disconnect, 54
 - getInstance, 54
 - getRequestTokens, 55
 - getSpaceTokens, 55
 - getURLs, 55
 - getVersion, 56
 - implementation, 59
 - info, 56

- logger, 59
- mkDir, 56
- ping, 57
- putURLs, 57
- release, 57
- releaseGet, 58
- releasePut, 58
- remove, 58
- request_timeout, 59
- requestBringOnline, 58
- requestBringOnlineStatus, 59
- service_endpoint, 60
- Timeout, 59
- version, 60
- SRMClientRequest, 60
 - file_ids, 61
 - finished_success, 61
 - long_list, 61
 - request_id, 61
 - request_token, 61
 - space_token, 61
 - SRMClientRequest, 61
 - surl_failures, 61
 - surl_statuses, 61
 - surls, 62
 - waiting_time, 62
- SRMFileInfo, 62
- SRMFileMetaData, 62
- SRMInfo, 62
- SRMInvalidRequestException, 63
- SRMURL, 63
 - BaseURL, 63
 - ContactURL, 63
 - Endpoint, 63
 - FileName, 63
 - FullURL, 64
 - PortDefined, 64
 - SetSRMVersion, 64
 - ShortURL, 64
 - SRMURL, 63
- ssl_
 - Arc::PayloadTLSStream, 43
- STACK_OF
 - Arc::PayloadTLSStream, 43
- stream_
 - Arc::PayloadHTTP, 39
- stream_own_
 - Arc::PayloadHTTP, 39
- surl_failures
 - SRMClientRequest, 61
- surl_statuses
 - SRMClientRequest, 61
- surls
 - SRMClientRequest, 62
- Timeout
 - SRMClient, 59
- transfer
 - Arc::HTTPSClientConnector, 26
 - Arc::HTTPSClientConnectorGlobus, 27
 - Arc::HTTPSClientConnectorGSSAPI, 28
- uri_
 - Arc::PayloadHTTP, 39
- version
 - SRMClient, 60
- version_major_
 - Arc::PayloadHTTP, 39
- version_minor_
 - Arc::PayloadHTTP, 39
- waiting_time
 - SRMClientRequest, 62
- XACMLCondition
 - ArcSec::XACMLCondition, 67
- XACMLEvaluationCtx
 - ArcSec::XACMLEvaluationCtx, 67
- XACMLPolicy
 - ArcSec::XACMLPolicy, 69
- XACMLTarget
 - ArcSec::XACMLTarget, 71