

## Hosting Environment (Daemon) Chain Components

Generated by Doxygen 1.7.5

Mon Mar 5 2012 16:29:15



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Data Structure Index</b>	<b>7</b>
3.1	Data Structures . . . . .	7
<b>4</b>	<b>Namespace Documentation</b>	<b>11</b>
4.1	ArcSec Namespace Reference . . . . .	11
4.1.1	Detailed Description . . . . .	13
4.1.2	Typedef Documentation . . . . .	13
4.1.2.1	AndList . . . . .	13
4.1.2.2	Match . . . . .	13
<b>5</b>	<b>Data Structure Documentation</b>	<b>15</b>
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 . . . . .	15
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	<a href="#">ArcSec::ArcAttributeProxy Class Reference</a>	16
5.4.1	<a href="#">Detailed Description</a>	17
5.5	<a href="#">ArcSec::ArcAuthZ Class Reference</a>	17
5.5.1	<a href="#">Detailed Description</a>	17
5.5.2	<a href="#">Member Function Documentation</a>	17
5.5.2.1	<a href="#">Handle</a>	17
5.5.2.2	<a href="#">MakePDPs</a>	17
5.6	<a href="#">ArcSec::ArcEvaluationCtx Class Reference</a>	18
5.6.1	<a href="#">Detailed Description</a>	18
5.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	18
5.6.2.1	<a href="#">ArcEvaluationCtx</a>	18
5.6.3	<a href="#">Member Function Documentation</a>	18
5.6.3.1	<a href="#">split</a>	18
5.7	<a href="#">ArcSec::ArcEvaluator Class Reference</a>	18
5.7.1	<a href="#">Detailed Description</a>	19
5.7.2	<a href="#">Member Function Documentation</a>	19
5.7.2.1	<a href="#">evaluate</a>	19
5.8	<a href="#">ArcSec::ArcFnFactory Class Reference</a>	19
5.8.1	<a href="#">Detailed Description</a>	19
5.8.2	<a href="#">Member Function Documentation</a>	19
5.8.2.1	<a href="#">createFn</a>	19
5.9	<a href="#">ArcSec::ArcPDP Class Reference</a>	19
5.9.1	<a href="#">Detailed Description</a>	20
5.10	<a href="#">ArcSec::ArcPolicy Class Reference</a>	20
5.10.1	<a href="#">Detailed Description</a>	20
5.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	20
5.10.2.1	<a href="#">ArcPolicy</a>	20
5.10.2.2	<a href="#">ArcPolicy</a>	20
5.10.2.3	<a href="#">ArcPolicy</a>	20
5.10.3	<a href="#">Member Function Documentation</a>	20
5.10.3.1	<a href="#">make_policy</a>	20
5.11	<a href="#">ArcSec::ArcRequest Class Reference</a>	21
5.12	<a href="#">ArcSec::ArcRequestItem Class Reference</a>	21
5.12.1	<a href="#">Detailed Description</a>	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	22
5.16	ArcSec::AttributeSelector Class Reference	22
5.17	ArcSHCLegacy::AuthUser Class Reference	22
5.18	ArcSHCLegacy::AuthVO Class Reference	22
5.19	Arc::DataPointGridFTP::CBArg Class Reference	22
5.20	ArcSHCLegacy::LegacyPDP::cfgfile Class Reference	23
5.21	ArcSHCLegacy::LegacyMap::cfgfile Class Reference	23
5.22	ArcSHCLegacy::ConfigParser Class Reference	23
5.23	ArcMCCTLS::ConfigTLSMCC Class Reference	23
5.24	Arc::DataPointARC Class Reference	23
5.24.1	Detailed Description	23
5.25	Arc::DataPointFile Class Reference	23
5.25.1	Detailed Description	24
5.26	Arc::DataPointGFAL Class Reference	24
5.26.1	Detailed Description	24
5.27	Arc::DataPointGridFTP Class Reference	24
5.27.1	Detailed Description	24
5.28	Arc::DataPointHTTP Class Reference	25
5.28.1	Detailed Description	25
5.29	Arc::DataPointLDAP Class Reference	25
5.29.1	Detailed Description	25
5.30	Arc::DataPointLFC Class Reference	25
5.30.1	Detailed Description	25
5.31	Arc::DataPointRLS Class Reference	26
5.31.1	Detailed Description	26
5.32	Arc::DataPointSRM Class Reference	26
5.32.1	Detailed Description	26
5.33	Arc::DataPointXrootd Class Reference	26
5.33.1	Detailed Description	27
5.34	ArcMCCTLSec::DelegationCollector Class Reference	27

5.35	<a href="#">ArcMCCTLSec::DelegationMultiSecAttr Class Reference</a>	27
5.36	<a href="#">ArcSec::DelegationPDP Class Reference</a>	27
5.36.1	<a href="#">Detailed Description</a>	27
5.37	<a href="#">ArcMCCTLSec::DelegationSecAttr Class Reference</a>	27
5.38	<a href="#">ArcSec::DelegationSH Class Reference</a>	28
5.39	<a href="#">ArcSec::DenyPDP Class Reference</a>	28
5.39.1	<a href="#">Detailed Description</a>	28
5.40	<a href="#">ArcSec::GACLEvaluator Class Reference</a>	28
5.40.1	<a href="#">Member Function Documentation</a>	28
5.40.1.1	<a href="#">evaluate</a>	28
5.41	<a href="#">ArcSec::GACLPDP Class Reference</a>	28
5.42	<a href="#">ArcSec::GACLPolicy Class Reference</a>	29
5.43	<a href="#">ArcSec::GACLRequest Class Reference</a>	29
5.44	<a href="#">ArcSHCLegacy::AuthUser::group_t Class Reference</a>	29
5.45	<a href="#">Arc::LDAPQuery Class Reference</a>	29
5.45.1	<a href="#">Detailed Description</a>	29
5.45.2	<a href="#">Constructor &amp; Destructor Documentation</a>	29
5.45.2.1	<a href="#">LDAPQuery</a>	30
5.45.2.2	<a href="#">~LDAPQuery</a>	30
5.45.3	<a href="#">Member Function Documentation</a>	30
5.45.3.1	<a href="#">Query</a>	30
5.45.3.2	<a href="#">Result</a>	30
5.46	<a href="#">ArcSHCLegacy::LegacyMap Class Reference</a>	30
5.47	<a href="#">ArcSHCLegacy::LegacyPDP Class Reference</a>	30
5.48	<a href="#">ArcSHCLegacy::LegacySecAttr Class Reference</a>	31
5.49	<a href="#">ArcSHCLegacy::LegacySecHandler Class Reference</a>	31
5.50	<a href="#">Arc::Lister Class Reference</a>	31
5.51	<a href="#">ArcMCCGSI::MCC_GSI_Client Class Reference</a>	31
5.52	<a href="#">ArcMCCGSI::MCC_GSI_Service Class Reference</a>	31
5.53	<a href="#">ArcMCCHTTP::MCC_HTTP Class Reference</a>	31
5.53.1	<a href="#">Detailed Description</a>	32
5.54	<a href="#">ArcMCCHTTP::MCC_HTTP_Client Class Reference</a>	32
5.54.1	<a href="#">Detailed Description</a>	32
5.55	<a href="#">ArcMCCHTTP::MCC_HTTP_Service Class Reference</a>	33

5.55.1 Detailed Description . . . . .	33
5.56 ArcMCCMsgValidator::MCC_MsgValidator Class Reference . . . . .	33
5.57 ArcMCCMsgValidator::MCC_MsgValidator_Service Class Reference . . . . .	34
5.58 ArcMCCSOAP::MCC_SOAP Class Reference . . . . .	34
5.58.1 Detailed Description . . . . .	34
5.59 ArcMCCSOAP::MCC_SOAP_Client Class Reference . . . . .	34
5.60 ArcMCCSOAP::MCC_SOAP_Service Class Reference . . . . .	35
5.60.1 Detailed Description . . . . .	35
5.61 ArcMCCTCP::MCC_TCP Class Reference . . . . .	35
5.61.1 Detailed Description . . . . .	36
5.62 ArcMCCTCP::MCC_TCP_Client Class Reference . . . . .	36
5.62.1 Detailed Description . . . . .	36
5.63 ArcMCCTCP::MCC_TCP_Service::mcc_tcp_exec_t Class Reference . . . . .	36
5.64 ArcMCCTCP::MCC_TCP_Service::mcc_tcp_handle_t Class Reference . . . . .	36
5.65 ArcMCCTCP::MCC_TCP_Service Class Reference . . . . .	37
5.65.1 Detailed Description . . . . .	37
5.65.2 Constructor & Destructor Documentation . . . . .	37
5.65.2.1 MCC_TCP_Service . . . . .	37
5.66 ArcMCCTLS::MCC_TLS Class Reference . . . . .	38
5.66.1 Detailed Description . . . . .	38
5.67 ArcMCCTLS::MCC_TLS_Client Class Reference . . . . .	38
5.67.1 Detailed Description . . . . .	38
5.68 ArcMCCTLS::MCC_TLS_Service Class Reference . . . . .	39
5.68.1 Detailed Description . . . . .	39
5.69 ArcMCCGSI::PayloadGSIStream Class Reference . . . . .	39
5.70 ArcMCCHTTP::PayloadHTTP Class Reference . . . . .	39
5.70.1 Detailed Description . . . . .	40
5.70.2 Constructor & Destructor Documentation . . . . .	41
5.70.2.1 PayloadHTTP . . . . .	41
5.70.2.2 PayloadHTTP . . . . .	41
5.70.2.3 PayloadHTTP . . . . .	41
5.70.2.4 PayloadHTTP . . . . .	41
5.70.2.5 PayloadHTTP . . . . .	41
5.70.3 Member Function Documentation . . . . .	41

5.70.3.1	Attribute	41
5.70.3.2	Attribute	41
5.70.3.3	Attributes	42
5.70.3.4	Body	42
5.70.3.5	Flush	42
5.70.3.6	get_body	42
5.70.3.7	read	42
5.70.3.8	read_header	42
5.70.3.9	readline	42
5.70.4	Field Documentation	42
5.70.4.1	attributes_	42
5.70.4.2	body_own_	43
5.70.4.3	chunk_size_	43
5.70.4.4	chunked_	43
5.70.4.5	code_	43
5.70.4.6	end_	43
5.70.4.7	length_	43
5.70.4.8	method_	43
5.70.4.9	rbody_	43
5.70.4.10	reason_	43
5.70.4.11	sbody_	43
5.70.4.12	stream_	44
5.70.4.13	stream_own_	44
5.70.4.14	uri_	44
5.70.4.15	version_major_	44
5.70.4.16	version_minor_	44
5.71	ArcMCCTCP::PayloadTCPSocket Class Reference	44
5.71.1	Detailed Description	44
5.71.2	Constructor & Destructor Documentation	45
5.71.2.1	PayloadTCPSocket	45
5.71.2.2	PayloadTCPSocket	45
5.71.2.3	PayloadTCPSocket	45
5.71.2.4	PayloadTCPSocket	45
5.71.2.5	PayloadTCPSocket	45



5.72 ArcMCCTLS::PayloadTLMCC Class Reference . . . . .	45
5.72.1 Constructor & Destructor Documentation . . . . .	46
5.72.1.1 PayloadTLMCC . . . . .	46
5.72.1.2 PayloadTLMCC . . . . .	46
5.72.1.3 PayloadTLMCC . . . . .	46
5.73 ArcMCCTLS::PayloadTLSStream Class Reference . . . . .	46
5.73.1 Detailed Description . . . . .	47
5.73.2 Constructor & Destructor Documentation . . . . .	47
5.73.2.1 PayloadTLSStream . . . . .	47
5.73.2.2 ~PayloadTLSStream . . . . .	47
5.73.3 Member Function Documentation . . . . .	47
5.73.3.1 GetCert . . . . .	47
5.73.3.2 GetPeerCert . . . . .	47
5.73.3.3 STACK_OF . . . . .	48
5.73.4 Field Documentation . . . . .	48
5.73.4.1 ssl_ . . . . .	48
5.74 ArcSec::ArcAuthZ::PDPDesc Class Reference . . . . .	48
5.75 ArcSec::PDPServiceInvoker Class Reference . . . . .	48
5.75.1 Detailed Description . . . . .	48
5.76 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference . . . . .	48
5.76.1 Detailed Description . . . . .	48
5.77 ArcSec::SAMLTokenSH Class Reference . . . . .	49
5.77.1 Detailed Description . . . . .	49
5.78 ArcSec::SimpleListPDP Class Reference . . . . .	49
5.78.1 Detailed Description . . . . .	49
5.79 ArcSHCLegacy::SimpleMap Class Reference . . . . .	49
5.80 ArcSHCLegacy::AuthUser::source_t Struct Reference . . . . .	49
5.81 ArcSHCLegacy::UnixMap::source_t Struct Reference . . . . .	50
5.82 Arc::SRM1Client Class Reference . . . . .	50
5.82.1 Member Function Documentation . . . . .	51
5.82.1.1 abort . . . . .	51
5.82.1.2 checkPermissions . . . . .	51
5.82.1.3 copy . . . . .	51
5.82.1.4 getRequestTokens . . . . .	52

5.82.1.5	<a href="#">getSpaceTokens</a>	52
5.82.1.6	<a href="#">getURLs</a>	52
5.82.1.7	<a href="#">getURLsStatus</a>	53
5.82.1.8	<a href="#">info</a>	53
5.82.1.9	<a href="#">info</a>	54
5.82.1.10	<a href="#">mkDir</a>	54
5.82.1.11	<a href="#">ping</a>	54
5.82.1.12	<a href="#">putURLs</a>	55
5.82.1.13	<a href="#">putURLsStatus</a>	55
5.82.1.14	<a href="#">release</a>	56
5.82.1.15	<a href="#">releaseGet</a>	56
5.82.1.16	<a href="#">releasePut</a>	56
5.82.1.17	<a href="#">remove</a>	57
5.82.1.18	<a href="#">requestBringOnline</a>	57
5.82.1.19	<a href="#">requestBringOnlineStatus</a>	57
5.83	<a href="#">Arc::SRM22Client Class Reference</a>	58
5.83.1	<a href="#">Constructor &amp; Destructor Documentation</a>	59
5.83.1.1	<a href="#">SRM22Client</a>	59
5.83.1.2	<a href="#">~SRM22Client</a>	59
5.83.2	<a href="#">Member Function Documentation</a>	59
5.83.2.1	<a href="#">abort</a>	59
5.83.2.2	<a href="#">checkPermissions</a>	59
5.83.2.3	<a href="#">copy</a>	59
5.83.2.4	<a href="#">getRequestTokens</a>	60
5.83.2.5	<a href="#">getSpaceTokens</a>	60
5.83.2.6	<a href="#">getURLs</a>	60
5.83.2.7	<a href="#">getURLsStatus</a>	60
5.83.2.8	<a href="#">info</a>	60
5.83.2.9	<a href="#">info</a>	60
5.83.2.10	<a href="#">mkDir</a>	61
5.83.2.11	<a href="#">ping</a>	61
5.83.2.12	<a href="#">putURLs</a>	61
5.83.2.13	<a href="#">putURLsStatus</a>	61
5.83.2.14	<a href="#">release</a>	61

5.83.2.15	releaseGet	61
5.83.2.16	releasePut	62
5.83.2.17	remove	62
5.83.2.18	requestBringOnline	62
5.83.2.19	requestBringOnlineStatus	62
5.84	Arc::SRMClient Class Reference	62
5.84.1	Detailed Description	64
5.84.2	Constructor & Destructor Documentation	64
5.84.2.1	~SRMClient	64
5.84.3	Member Function Documentation	64
5.84.3.1	abort	64
5.84.3.2	checkPermissions	64
5.84.3.3	copy	65
5.84.3.4	getInstance	65
5.84.3.5	getRequestTokens	65
5.84.3.6	getSpaceTokens	66
5.84.3.7	getTURLs	66
5.84.3.8	getTURLsStatus	67
5.84.3.9	getVersion	67
5.84.3.10	info	67
5.84.3.11	info	68
5.84.3.12	mkDir	68
5.84.3.13	ping	68
5.84.3.14	putTURLs	69
5.84.3.15	putTURLsStatus	69
5.84.3.16	release	70
5.84.3.17	releaseGet	70
5.84.3.18	releasePut	70
5.84.3.19	remove	71
5.84.3.20	requestBringOnline	71
5.84.3.21	requestBringOnlineStatus	71
5.84.4	Field Documentation	72
5.84.4.1	service_endpoint	72
5.85	Arc::SRMClientRequest Class Reference	72

5.85.1 Detailed Description . . . . .	72
5.85.2 Constructor & Destructor Documentation . . . . .	73
5.85.2.1 SRMClientRequest . . . . .	73
5.85.2.2 SRMClientRequest . . . . .	73
5.85.3 Member Function Documentation . . . . .	73
5.85.3.1 waiting_time . . . . .	73
5.86 SRMFileInfo Class Reference . . . . .	73
5.86.1 Detailed Description . . . . .	73
5.87 Arc::SRMFileMetaData Struct Reference . . . . .	73
5.87.1 Detailed Description . . . . .	74
5.88 SRMInfo Class Reference . . . . .	74
5.88.1 Detailed Description . . . . .	74
5.89 Arc::SRMInvalidRequestException Class Reference . . . . .	74
5.89.1 Detailed Description . . . . .	74
5.90 SRMURL Class Reference . . . . .	74
5.90.1 Constructor & Destructor Documentation . . . . .	75
5.90.1.1 SRMURL . . . . .	75
5.90.2 Member Function Documentation . . . . .	75
5.90.2.1 BaseURL . . . . .	75
5.90.2.2 ContactURL . . . . .	75
5.90.2.3 Endpoint . . . . .	75
5.90.2.4 FileName . . . . .	75
5.90.2.5 FullURL . . . . .	75
5.90.2.6 PortDefined . . . . .	75
5.90.2.7 SetSRMVersion . . . . .	75
5.90.2.8 ShortURL . . . . .	76
5.91 ArcSHCLegacy::UnixMap::unix_user_t Class Reference . . . . .	76
5.92 ArcSHCLegacy::UnixMap Class Reference . . . . .	76
5.93 ArcSec::UsernameTokenSH Class Reference . . . . .	76
5.93.1 Detailed Description . . . . .	76
5.94 ArcSHCLegacy::voms Struct Reference . . . . .	76
5.94.1 Detailed Description . . . . .	77
5.94.2 Field Documentation . . . . .	77
5.94.2.1 attrs . . . . .	77

5.94.2.2	server	77
5.94.2.3	voname	77
5.95	ArcSHCLegacy::voms_attrs Struct Reference	77
5.95.1	Detailed Description	77
5.95.2	Field Documentation	78
5.95.2.1	cap	78
5.95.2.2	group	78
5.95.2.3	role	78
5.96	ArcSec::X509TokenSH Class Reference	78
5.96.1	Detailed Description	78
5.97	ArcSec::XACMLAlgFactory Class Reference	78
5.97.1	Detailed Description	78
5.97.2	Member Function Documentation	79
5.97.2.1	createAlg	79
5.98	ArcSec::XACMLApply Class Reference	79
5.99	ArcSec::XACMLAttributeFactory Class Reference	79
5.99.1	Detailed Description	79
5.99.2	Member Function Documentation	79
5.99.2.1	createValue	79
5.100	ArcSec::XACMLAttributeProxy Class Reference	80
5.100.1	Detailed Description	80
5.101	ArcSec::XACMLCondition Class Reference	80
5.101.1	Detailed Description	80
5.101.2	Constructor & Destructor Documentation	80
5.101.2.1	XACMLCondition	80
5.102	ArcSec::XACMLEvaluationCtx Class Reference	80
5.102.1	Detailed Description	81
5.102.2	Constructor & Destructor Documentation	81
5.102.2.1	XACMLEvaluationCtx	81
5.103	ArcSec::XACMLEvaluator Class Reference	81
5.103.1	Detailed Description	81
5.103.2	Member Function Documentation	81
5.103.2.1	evaluate	81
5.104	ArcSec::XACMLFnFactory Class Reference	82

5.104.1 Detailed Description . . . . .	82
5.104.2 Member Function Documentation . . . . .	82
5.104.2.1 createFn . . . . .	82
5.105ArcSec::XACMLPDP Class Reference . . . . .	82
5.105.1 Detailed Description . . . . .	82
5.106ArcSec::XACMLPolicy Class Reference . . . . .	82
5.106.1 Detailed Description . . . . .	83
5.106.2 Constructor & Destructor Documentation . . . . .	83
5.106.2.1 XACMLPolicy . . . . .	83
5.106.2.2 XACMLPolicy . . . . .	83
5.106.2.3 XACMLPolicy . . . . .	83
5.106.3 Member Function Documentation . . . . .	83
5.106.3.1 make_policy . . . . .	83
5.107ArcSec::XACMLRequest Class Reference . . . . .	83
5.107.1 Member Function Documentation . . . . .	84
5.107.1.1 getEvalName . . . . .	84
5.107.1.2 getName . . . . .	84
5.108ArcSec::XACMLRule Class Reference . . . . .	84
5.108.1 Detailed Description . . . . .	84
5.109ArcSec::XACMLTarget Class Reference . . . . .	84
5.109.1 Detailed Description . . . . .	84
5.109.2 Constructor & Destructor Documentation . . . . .	85
5.109.2.1 XACMLTarget . . . . .	85
5.110ArcSec::XACMLTargetMatch Class Reference . . . . .	85
5.111ArcSec::XACMLTargetMatchGroup Class Reference . . . . .	85
5.112ArcSec::XACMLTargetSection Class Reference . . . . .	85

# Chapter 1

## Namespace Index

### 1.1 Namespace List

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

<a href="#">ArcSec</a>	
<a href="#">ArcRequest</a> , 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 . . . . .	16
ArcSec::ArcAuthZ . . . . .	17
ArcSec::ArcEvaluationCtx . . . . .	18
ArcSec::ArcEvaluator . . . . .	18
ArcSec::ArcFnFactory . . . . .	19
ArcSec::ArcPDP . . . . .	19
ArcSec::ArcPolicy . . . . .	20
ArcSec::ArcRequest . . . . .	21
ArcSec::ArcRequestItem . . . . .	21
ArcSec::ArcRequestTuple . . . . .	21
ArcSec::ArcRule . . . . .	21
ArcSec::AttributeDesignator . . . . .	22
ArcSec::AttributeSelector . . . . .	22
ArcSHCLegacy::AuthUser . . . . .	22
ArcSHCLegacy::AuthVO . . . . .	22
Arc::DataPointGridFTP::CBArg . . . . .	22
ArcSHCLegacy::LegacyPDP::cfgfile . . . . .	23
ArcSHCLegacy::LegacyMap::cfgfile . . . . .	23
ArcSHCLegacy::ConfigParser . . . . .	23
ArcMCCTLS::ConfigTLMCC . . . . .	23
Arc::DataPointARC . . . . .	23
Arc::DataPointFile . . . . .	23
Arc::DataPointGFAL . . . . .	24
Arc::DataPointGridFTP . . . . .	24
Arc::DataPointHTTP . . . . .	25
Arc::DataPointLDAP . . . . .	25

Arc::DataPointLFC . . . . .	25
Arc::DataPointRLS . . . . .	26
Arc::DataPointSRM . . . . .	26
Arc::DataPointXrootd . . . . .	26
ArcMCCTLSec::DelegationCollector . . . . .	27
ArcMCCTLSec::DelegationMultiSecAttr . . . . .	27
ArcSec::DelegationPDP . . . . .	27
ArcMCCTLSec::DelegationSecAttr . . . . .	27
ArcSec::DelegationSH . . . . .	28
ArcSec::DenyPDP . . . . .	28
ArcSec::GACLEvaluator . . . . .	28
ArcSec::GACLPDP . . . . .	28
ArcSec::GACLPolicy . . . . .	29
ArcSec::GACLRequest . . . . .	29
ArcSHCLegacy::AuthUser::group_t . . . . .	29
Arc::LDAPQuery . . . . .	29
ArcSHCLegacy::LegacyMap . . . . .	30
ArcSHCLegacy::LegacyPDP . . . . .	30
ArcSHCLegacy::LegacySecAttr . . . . .	31
ArcSHCLegacy::LegacySecHandler . . . . .	31
Arc::Lister . . . . .	31
ArcMCCGSI::MCC_GSI_Client . . . . .	31
ArcMCCGSI::MCC_GSI_Service . . . . .	31
ArcMCCHTTP::MCC_HTTP . . . . .	31
ArcMCCHTTP::MCC_HTTP_Client . . . . .	32
ArcMCCHTTP::MCC_HTTP_Service . . . . .	33
ArcMCCMsgValidator::MCC_MsgValidator . . . . .	33
ArcMCCMsgValidator::MCC_MsgValidator_Service . . . . .	34
ArcMCCSOAP::MCC_SOAP . . . . .	34
ArcMCCSOAP::MCC_SOAP_Client . . . . .	34
ArcMCCSOAP::MCC_SOAP_Service . . . . .	35
ArcMCCTCP::MCC_TCP . . . . .	35
ArcMCCTCP::MCC_TCP_Client . . . . .	36
ArcMCCTCP::MCC_TCP_Service . . . . .	37
ArcMCCTCP::MCC_TCP_Service::mcc_tcp_exec_t . . . . .	36
ArcMCCTCP::MCC_TCP_Service::mcc_tcp_handle_t . . . . .	36
ArcMCCTLS::MCC_TLS . . . . .	38
ArcMCCTLS::MCC_TLS_Client . . . . .	38
ArcMCCTLS::MCC_TLS_Service . . . . .	39
ArcMCCGSI::PayloadGSISStream . . . . .	39
ArcMCCHTTP::PayloadHTTP . . . . .	39
ArcMCCTCP::PayloadTCPSocket . . . . .	44
ArcMCCTLS::PayloadTLSStream . . . . .	46
ArcMCCTLS::PayloadTLSMCC . . . . .	45
ArcSec::ArcAuthZ::PDPDesc . . . . .	48
ArcSec::PDPServiceInvoker . . . . .	48
ArcSec::SAML2SSO_AssertionConsumerSH . . . . .	48
ArcSec::SAMLTokenSH . . . . .	49

ArcSec::SimpleListPDP . . . . .	49
ArcSHCLegacy::SimpleMap . . . . .	49
ArcSHCLegacy::AuthUser::source_t . . . . .	49
ArcSHCLegacy::UnixMap::source_t . . . . .	50
Arc::SRMClient . . . . .	62
Arc::SRM1Client . . . . .	50
Arc::SRM22Client . . . . .	58
Arc::SRMClientRequest . . . . .	72
SRMFileInfo . . . . .	73
Arc::SRMFileMetaData . . . . .	73
SRMInfo . . . . .	74
Arc::SRMInvalidRequestException . . . . .	74
SRMURL . . . . .	74
ArcSHCLegacy::UnixMap::unix_user_t . . . . .	76
ArcSHCLegacy::UnixMap . . . . .	76
ArcSec::UsernameTokenSH . . . . .	76
ArcSHCLegacy::voms . . . . .	76
ArcSHCLegacy::voms_attrs . . . . .	77
ArcSec::X509TokenSH . . . . .	78
ArcSec::XACMLAlgFactory . . . . .	78
ArcSec::XACMLApply . . . . .	79
ArcSec::XACMLAttributeFactory . . . . .	79
ArcSec::XACMLAttributeProxy . . . . .	80
ArcSec::XACMLCondition . . . . .	80
ArcSec::XACMLEvaluationCtx . . . . .	80
ArcSec::XACMLEvaluator . . . . .	81
ArcSec::XACMLFnFactory . . . . .	82
ArcSec::XACMLPDP . . . . .	82
ArcSec::XACMLPolicy . . . . .	82
ArcSec::XACMLRequest . . . . .	83
ArcSec::XACMLRule . . . . .	84
ArcSec::XACMLTarget . . . . .	84
ArcSec::XACMLTargetMatch . . . . .	85
ArcSec::XACMLTargetMatchGroup . . . . .	85
ArcSec::XACMLTargetSection . . . . .	85



## Chapter 3

# Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">ArcSec::AllowPDP</a>	
This PDP always return true (allow)	15
<a href="#">ArcSec::ArcAlgFactory</a>	
Algorithm factory class for Arc	15
<a href="#">ArcSec::ArcAttributeFactory</a>	
Attribute factory class for Arc specified attributes	16
<a href="#">ArcSec::ArcAttributeProxy</a>	
Arc specific AttributeProxy class	16
<a href="#">ArcSec::ArcAuthZ</a>	
Tests message against list of PDPs	17
<a href="#">ArcSec::ArcEvaluationCtx</a>	
EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc	18
<a href="#">ArcSec::ArcEvaluator</a>	
Execute the policy evaluation, based on the request and policy	18
<a href="#">ArcSec::ArcFnFactory</a>	
Function factory class for Arc specified attributes	19
<a href="#">ArcSec::ArcPDP</a>	
<a href="#">ArcPDP</a> - PDP which can handle the Arc specific request and policy schema	19
<a href="#">ArcSec::ArcPolicy</a>	
<a href="#">ArcPolicy</a> class to parse and operate Arc specific <Policy> node	20
<a href="#">ArcSec::ArcRequest</a>	21
<a href="#">ArcSec::ArcRequestItem</a>	
Container, <Subjects, Actions, Objects, Contexts> tuple	21
<a href="#">ArcSec::ArcRequestTuple</a>	
RequestTuple, container which includes the	21
<a href="#">ArcSec::ArcRule</a>	
<a href="#">ArcRule</a> class to parse Arc specific <Rule> node	21

ArcSec::AttributeDesignator	22
ArcSec::AttributeSelector	22
ArcSHCLegacy::AuthUser	22
ArcSHCLegacy::AuthVO	22
Arc::DataPointGridFTP::CBArg	22
ArcSHCLegacy::LegacyPDP::cfgfile	23
ArcSHCLegacy::LegacyMap::cfgfile	23
ArcSHCLegacy::ConfigParser	23
ArcMCCTLS::ConfigTLMCC	23
Arc::DataPointARC	23
Arc::DataPointFile	23
Arc::DataPointGFAL	24
Arc::DataPointGridFTP	24
Arc::DataPointHTTP	25
Arc::DataPointLDAP	25
Arc::DataPointLFC	25
Arc::DataPointRLS	26
Arc::DataPointSRM	26
Arc::DataPointXrootd	26
ArcMCCTLSec::DelegationCollector	27
ArcMCCTLSec::DelegationMultiSecAttr	27
ArcSec::DelegationPDP	27
ArcMCCTLSec::DelegationSecAttr	27
ArcSec::DelegationSH	28
ArcSec::DenyPDP	
This PDP always returns false (deny)	28
ArcSec::GACLEvaluator	28
ArcSec::GACLPDP	28
ArcSec::GACLPolicy	29
ArcSec::GACLRequest	29
ArcSHCLegacy::AuthUser::group_t	29
Arc::LDAPQuery	29
ArcSHCLegacy::LegacyMap	30
ArcSHCLegacy::LegacyPDP	30
ArcSHCLegacy::LegacySecAttr	31
ArcSHCLegacy::LegacySecHandler	31
Arc::Lister	31
ArcMCCGSI::MCC_GSI_Client	31
ArcMCCGSI::MCC_GSI_Service	31
ArcMCCHTTP::MCC_HTTP	
A base class for HTTP client and service MCCs	31
ArcMCCHTTP::MCC_HTTP_Client	32
ArcMCCHTTP::MCC_HTTP_Service	33
ArcMCCMsgValidator::MCC_MsgValidator	33
ArcMCCMsgValidator::MCC_MsgValidator_Service	34
ArcMCCSOAP::MCC_SOAP	
A base class for SOAP client and service MCCs	34
ArcMCCSOAP::MCC_SOAP_Client	34
ArcMCCSOAP::MCC_SOAP_Service	35

<a href="#">ArcMCCTCP::MCC_TCP</a>	
A base class for TCP client and service MCCs	35
<a href="#">ArcMCCTCP::MCC_TCP_Client</a>	36
<a href="#">ArcMCCTCP::MCC_TCP_Service::mcc_tcp_exec_t</a>	36
<a href="#">ArcMCCTCP::MCC_TCP_Service::mcc_tcp_handle_t</a>	36
<a href="#">ArcMCCTCP::MCC_TCP_Service</a>	37
<a href="#">ArcMCCTLS::MCC_TLS</a>	
A base class for TLS client and service MCCs	38
<a href="#">ArcMCCTLS::MCC_TLS_Client</a>	38
<a href="#">ArcMCCTLS::MCC_TLS_Service</a>	39
<a href="#">ArcMCCGSI::PayloadGSISStream</a>	39
<a href="#">ArcMCCHTTP::PayloadHTTP</a>	39
<a href="#">ArcMCCTCP::PayloadTCPsocket</a>	44
<a href="#">ArcMCCTLS::PayloadTSLMCC</a>	45
<a href="#">ArcMCCTLS::PayloadTLSSStream</a>	46
<a href="#">ArcSec::ArcAuthZ::PDPDesc</a>	48
<a href="#">ArcSec::PDPServletInvoker</a>	
PDPServletInvoker - client which will invoke pdpservice	48
<a href="#">ArcSec::SAML2SSO_AssertionConsumerSH</a>	
Implement the functionality of the Service Provider in SAML2 SSO profile	48
<a href="#">ArcSec::SAMLTokenSH</a>	
Adds WS-Security SAML Token into SOAP Header	49
<a href="#">ArcSec::SimpleListPDP</a>	
Tests X509 subject against list of subjects in file	49
<a href="#">ArcSHCLegacy::SimpleMap</a>	49
<a href="#">ArcSHCLegacy::AuthUser::source_t</a>	49
<a href="#">ArcSHCLegacy::UnixMap::source_t</a>	50
<a href="#">Arc::SRM1Client</a>	50
<a href="#">Arc::SRM22Client</a>	58
<a href="#">Arc::SRMClient</a>	62
<a href="#">Arc::SRMClientRequest</a>	
Class to represent a SRM request	72
<a href="#">SRMFileInfo</a>	73
<a href="#">Arc::SRMFileMetaData</a>	
SRM-related file metadata	73
<a href="#">SRMInfo</a>	74
<a href="#">Arc::SRMInvalidRequestException</a>	
General exception to represent a bad SRM request	74
<a href="#">SRMURL</a>	74
<a href="#">ArcSHCLegacy::UnixMap::unix_user_t</a>	76
<a href="#">ArcSHCLegacy::UnixMap</a>	76
<a href="#">ArcSec::UsernameTokenSH</a>	
Adds WS-Security Username Token into SOAP Header	76
<a href="#">ArcSHCLegacy::voms</a>	76
<a href="#">ArcSHCLegacy::voms_attrs</a>	77
<a href="#">ArcSec::X509TokenSH</a>	
Adds WS-Security X509 Token into SOAP Header	78
<a href="#">ArcSec::XACMLAlgFactory</a>	
Algorithm factory class for XACML	78

<a href="#">ArcSec::XACMLApply</a>	79
<a href="#">ArcSec::XACMLAttributeFactory</a>	
Attribute factory class for XACML specified attributes	79
<a href="#">ArcSec::XACMLAttributeProxy</a>	
XACML specific AttributeProxy class	80
<a href="#">ArcSec::XACMLCondition</a>	
XACMLCondition class to parse and operate XACML specific <- Condition> node	80
<a href="#">ArcSec::XACMLEvaluationCtx</a>	
EvaluationCtx, in charge of storing some context information for eval- uation, including Request, current time, etc	80
<a href="#">ArcSec::XACMLEvaluator</a>	
Execute the policy evaluation, based on the request and policy	81
<a href="#">ArcSec::XACMLFnFactory</a>	
Function factory class for XACML specified attributes	82
<a href="#">ArcSec::XACMLPDP</a>	
XACMLPDP - PDP which can handle the XACML specific request and policy schema	82
<a href="#">ArcSec::XACMLPolicy</a>	
XACMLPolicy class to parse and operate XACML specific <Policy> node	82
<a href="#">ArcSec::XACMLRequest</a>	83
<a href="#">ArcSec::XACMLRule</a>	
XACMLRule class to parse XACML specific <Rule> node	84
<a href="#">ArcSec::XACMLTarget</a>	
XACMLTarget class to parse and operate XACML specific <Target> node	84
<a href="#">ArcSec::XACMLTargetMatch</a>	85
<a href="#">ArcSec::XACMLTargetMatchGroup</a>	85
<a href="#">ArcSec::XACMLTargetSection</a>	85



## Chapter 4

# Namespace Documentation

### 4.1 ArcSec Namespace Reference

#### Data Structures

- class [AllowPDP](#)  
*This PDP always return true (allow)*
- class [ArcAuthZ](#)  
*Tests message against list of PDPs.*
- class [ArcAlgFactory](#)  
*Algorithm factory class for Arc.*
- class [ArcAttributeFactory](#)  
*Attribute factory class for Arc specified attributes.*
- class [ArcAttributeProxy](#)  
*Arc specific AttributeProxy class.*
- class [ArcRequestTuple](#)  
*RequestTuple, container which includes the.*
- class [ArcEvaluationCtx](#)  
*EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc.*
- class [ArcEvaluator](#)  
*Execute the policy evaluation, based on the request and policy.*
- class [ArcFnFactory](#)  
*Function factory class for Arc specified attributes.*
- class [ArcPDP](#)  
*ArcPDP - PDP which can handle the Arc specific request and policy schema.*
- class [ArcPolicy](#)  
*ArcPolicy class to parse and operate Arc specific <Policy> node.*
- class [ArcRequest](#)
- class [ArcRequestItem](#)

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

- class [ArcRule](#)

*ArcRule class to parse Arc specific <Rule> node.*

- class [DelegationPDP](#)
- class [DelegationSH](#)
- class [DenyPDP](#)

*This PDP always returns false (deny)*

- class [GACLEvaluator](#)
- class [GACLPDP](#)
- class [GACLPolicy](#)
- class [GACLRequest](#)
- class [PDPServiceInvoker](#)

*PDPServiceInvoker - client which will invoke pdpservice.*

- class [SAML2SSO\\_AssertionConsumerSH](#)

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

- class [SAMLTokenSH](#)

*Adds WS-Security SAML Token into SOAP Header.*

- class [SimpleListPDP](#)

*Tests X509 subject against list of subjects in file.*

- class [UsernameTokenSH](#)

*Adds WS-Security Username Token into SOAP Header.*

- class [X509TokenSH](#)

*Adds WS-Security X509 Token into SOAP Header.*

- class [AttributeDesignator](#)
- class [AttributeSelector](#)
- class [XACMLAlgFactory](#)

*Algorithm factory class for XACML.*

- class [XACMLApply](#)
- class [XACMLAttributeFactory](#)

*Attribute factory class for XACML specified attributes.*

- class [XACMLAttributeProxy](#)

*XACML specific AttributeProxy class.*

- class [XACMLCondition](#)

*XACMLCondition class to parse and operate XACML specific <Condition> node.*

- class [XACMLEvaluationCtx](#)

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

- class [XACMLEvaluator](#)

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

- class [XACMLFnFactory](#)

*Function factory class for XACML specified attributes.*

- class [XACMLPDP](#)

*XACMLPDP - PDP which can handle the XACML specific request and policy schema.*

- class [XACMLPolicy](#)

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

- class [XACMLRequest](#)
- class [XACMLRule](#)

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

- class [XACMLTargetMatch](#)
- class [XACMLTargetMatchGroup](#)
- class [XACMLTargetSection](#)
- class [XACMLTarget](#)

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

## Typedefs

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

### 4.1.1 Detailed Description

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

### 4.1.2 Typedef Documentation

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

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

"And" relationship means the request should satisfy all of the items <Subject> <-SubFraction type="X500DN">/O=Grid/OU=KnowARC/CN=XYZ</SubFraction> <-SubFraction type="ShibName">urn:mace:shibboleth: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> <Sub-
Fraction type="string">urn:mace:shibboleth:examples</SubFraction> </Subject>
<GroupldRef location="/subjectgroup.xml">subgrpexample1</GroupldRef> </-
Subjects>
```

## Chapter 5

# Data Structure Documentation

### 5.1 ArcSec::AllowPDP Class Reference

```
#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

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

#### Public Member Functions

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

#### 5.2.1 Detailed Description

Algorithm factory class for Arc.

#### 5.2.2 Member Function Documentation

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

return a Alg object according to the "CombiningAlg" attribute in the <Policy> node; The [ArcAlgFactory](#) itself will release the Alg objects

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

- [ArcAlgFactory.h](#)

### 5.3 ArcSec::ArcAttributeFactory Class Reference

```
#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 Class Reference

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

#### Public Member Functions

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

### 5.4.1 Detailed Description

Arc specific AttributeProxy class.

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

- [ArcAttributeProxy.h](#)

## 5.5 ArcSec::ArcAuthZ Class Reference

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

### Data Structures

- class [PDPDesc](#)

### Public Member Functions

- virtual bool [Handle](#) (Arc::Message \*msg) const

### Protected Member Functions

- bool [MakePDPs](#) (Arc::XMLNode cfg)

### 5.5.1 Detailed Description

Tests message against list of PDPs.

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

### 5.5.2 Member Function Documentation

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

Get authorization decision

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

Create PDP according to conf info

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

- [ArcAuthZ.h](#)

## 5.6 ArcSec::ArcEvaluationCtx Class Reference

```
#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

```
#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

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

### Public Member Functions

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

### 5.8.1 Detailed Description

Function factory class for Arc specified attributes.

### 5.8.2 Member Function Documentation

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

return a Function object according to the "Function" attribute in the XML node; The [ArcFnFactory](#) itself will release the Function objects

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

- ArcFnFactory.h

## 5.9 ArcSec::ArcPDP Class Reference

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

### 5.9.1 Detailed Description

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

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

- [ArcPDP.h](#)

## 5.10 ArcSec::ArcPolicy Class Reference

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

### Public Member Functions

- [ArcPolicy](#) (void)
- [ArcPolicy](#) (const Arc::XMLNode node)
- [ArcPolicy](#) (const Arc::XMLNode node, EvaluatorContext \*ctx)
- virtual void [make\\_policy](#) ()

### 5.10.1 Detailed Description

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

### 5.10.2 Constructor & Destructor Documentation

#### 5.10.2.1 ArcSec::ArcPolicy::ArcPolicy ( void )

Constructor

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

Constructor

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

Constructor

### 5.10.3 Member Function Documentation

#### 5.10.3.1 virtual void ArcSec::ArcPolicy::make\_policy ( ) [virtual]

Parse XMLNode, and construct the low-level Rule object

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

- ArcPolicy.h

## 5.11 ArcSec::ArcRequest Class Reference

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

- ArcRequest.h

## 5.12 ArcSec::ArcRequestItem Class Reference

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

### 5.12.1 Detailed Description

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

Specified [ArcRequestItem](#) 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

```
#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

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

### 5.14.1 Detailed Description

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

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

- [ArcRule.h](#)

### 5.15 ArcSec::AttributeDesignator Class Reference

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

- [AttributeDesignator.h](#)

### 5.16 ArcSec::AttributeSelector Class Reference

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

- [AttributeSelector.h](#)

### 5.17 ArcSHCLegacy::AuthUser Class Reference

#### Data Structures

- class [group\\_t](#)
- struct [source\\_t](#)

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

- [auth.h](#)

### 5.18 ArcSHCLegacy::AuthVO Class Reference

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

- [auth.h](#)

### 5.19 Arc::DataPointGridFTP::CBArg Class Reference

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

- [DataPointGridFTP.h](#)

## 5.20 ArcSHCLegacy::LegacyPDP::cfgfile Class Reference

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

- LegacyPDP.h

## 5.21 ArcSHCLegacy::LegacyMap::cfgfile Class Reference

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

- LegacyMap.h

## 5.22 ArcSHCLegacy::ConfigParser Class Reference

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

- ConfigParser.h

## 5.23 ArcMCCTLS::ConfigTLMCC Class Reference

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

- ConfigTLMCC.h

## 5.24 Arc::DataPointARC Class Reference

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

### 5.24.1 Detailed Description

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

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

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

- DataPointARC.h

## 5.25 Arc::DataPointFile Class Reference

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

### 5.25.1 Detailed Description

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

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

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

- `DataPointFile.h`

## 5.26 Arc::DataPointGFAL Class Reference

```
#include <DataPointGFAL.h>
```

### 5.26.1 Detailed Description

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

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

- `DataPointGFAL.h`

## 5.27 Arc::DataPointGridFTP Class Reference

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

### Data Structures

- class [CBAArg](#)

### 5.27.1 Detailed Description

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

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

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

- `DataPointGridFTP.h`

## 5.28 Arc::DataPointHTTP Class Reference

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

### 5.28.1 Detailed Description

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

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

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

- DataPointHTTP.h

## 5.29 Arc::DataPointLDAP Class Reference

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

### 5.29.1 Detailed Description

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

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

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

- DataPointLDAP.h

## 5.30 Arc::DataPointLFC Class Reference

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

### 5.30.1 Detailed Description

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

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

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

- DataPointLFC.h

## 5.31 Arc::DataPointRLS Class Reference

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

### 5.31.1 Detailed Description

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

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

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

- DataPointRLS.h

## 5.32 Arc::DataPointSRM Class Reference

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

### 5.32.1 Detailed Description

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

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

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

- DataPointSRM.h

## 5.33 Arc::DataPointXrootd Class Reference

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



### 5.33.1 Detailed Description

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

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

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

- DataPointXrootd.h

## 5.34 ArcMCCTLSec::DelegationCollector Class Reference

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

- DelegationCollector.h

## 5.35 ArcMCCTLSec::DelegationMultiSecAttr Class Reference

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

- DelegationSecAttr.h

## 5.36 ArcSec::DelegationPDP Class Reference

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

### 5.36.1 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

## 5.37 ArcMCCTLSec::DelegationSecAttr Class Reference

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

- DelegationSecAttr.h

### 5.38 ArcSec::DelegationSH Class Reference

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

- DelegationSH.h

### 5.39 ArcSec::DenyPDP Class Reference

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

#### 5.39.1 Detailed Description

This PDP always returns false (deny)

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

- DenyPDP.h

### 5.40 ArcSec::GACLEvaluator Class Reference

#### Public Member Functions

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

#### 5.40.1 Member Function Documentation

5.40.1.1 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

### 5.41 ArcSec::GACLPDP Class Reference

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

- GACLPDP.h

## 5.42 ArcSec::GACLPolicy Class Reference

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

- GACLPolicy.h

## 5.43 ArcSec::GACLRequest Class Reference

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

- GACLRequest.h

## 5.44 ArcSHCLegacy::AuthUser::group\_t Class Reference

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

- auth.h

## 5.45 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)

### 5.45.1 Detailed Description

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

### 5.45.2 Constructor & Destructor Documentation

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

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

5.45.2.2 `Arc::LDAPQuery::~~LDAPQuery ( )`

Destructor. Will disconnect from the ldapserver if still connected.

### 5.45.3 Member Function Documentation

5.45.3.1 `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.

5.45.3.2 `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`

## 5.46 ArcSHCLegacy::LegacyMap Class Reference

### Data Structures

- class [cfgfile](#)

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

- `LegacyMap.h`

## 5.47 ArcSHCLegacy::LegacyPDP Class Reference

### Data Structures

- class [cfgfile](#)

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

- LegacyPDP.h

## 5.48 ArcSHCLegacy::LegacySecAttr Class Reference

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

- LegacySecAttr.h

## 5.49 ArcSHCLegacy::LegacySecHandler Class Reference

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

- LegacySecHandler.h

## 5.50 Arc::Lister Class Reference

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

- Lister.h

## 5.51 ArcMCCGSI::MCC\_GSI\_Client Class Reference

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

- MCCGSI.h

## 5.52 ArcMCCGSI::MCC\_GSI\_Service Class Reference

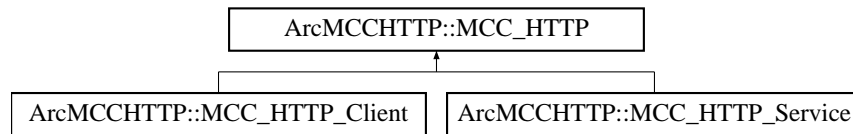
The documentation for this class was generated from the following file:

- MCCGSI.h

## 5.53 ArcMCCHTTP::MCC\_HTTP Class Reference

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

Inheritance diagram for ArcMCCHTTP::MCC\_HTTP:



### 5.53.1 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.

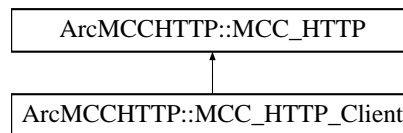
The documentation for this class was generated from the following file:

- MCCHTTP.h

## 5.54 ArcMCCHTTP::MCC\_HTTP\_Client Class Reference

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

Inheritance diagram for ArcMCCHTTP::MCC\_HTTP\_Client:



### 5.54.1 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:METHORD and HTTP:ENDPOINT specify method and URL in HTTP request. If not present meathod and URL are taken from configuration. In output/response message following attributes are present: HTTP:CODE - response code of HTTP HTTP:REASON - reason string of HTTP response HTTP:name - all 'name' attributes of HTTP header.

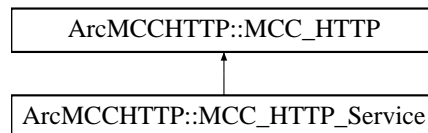
The documentation for this class was generated from the following file:

- MCCHTTP.h

## 5.55 ArcMCCHTTP::MCC\_HTTP\_Service Class Reference

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

Inheritance diagram for ArcMCCHTTP::MCC\_HTTP\_Service:



### 5.55.1 Detailed Description

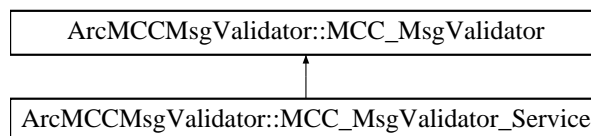
This class implements MCC to processes HTTP request. On input payload with - PayloadStreamInterface is expected. HTTP message is read from stream and its body is converted into PayloadRaw and passed to next MCC. Returned payload of PayloadRawInterface type is treated as body part of returning [PayloadHTTP](#).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

## 5.56 ArcMCCMsgValidator::MCC\_MsgValidator Class Reference

Inheritance diagram for ArcMCCMsgValidator::MCC\_MsgValidator:

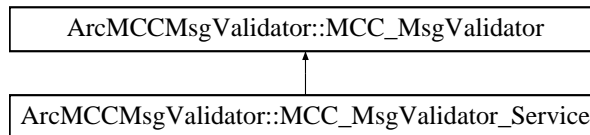


The documentation for this class was generated from the following file:

- MCCMsgValidator.h

## 5.57 ArcMCCMsgValidator::MCC\_MsgValidator\_Service Class - Reference

Inheritance diagram for ArcMCCMsgValidator::MCC\_MsgValidator\_Service:



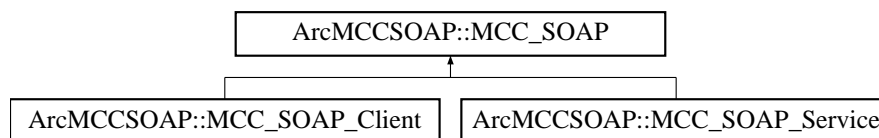
The documentation for this class was generated from the following file:

- MCCMsgValidator.h

## 5.58 ArcMCCSOAP::MCC\_SOAP Class Reference

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

Inheritance diagram for ArcMCCSOAP::MCC\_SOAP:



### 5.58.1 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.

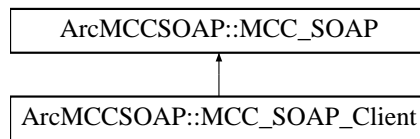
The documentation for this class was generated from the following file:

- MCCSOAP.h

## 5.59 ArcMCCSOAP::MCC\_SOAP\_Client Class Reference

Inheritance diagram for ArcMCCSOAP::MCC\_SOAP\_Client:





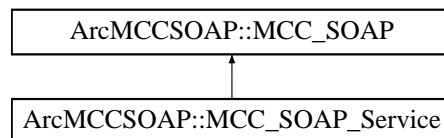
The documentation for this class was generated from the following file:

- MCCSOAP.h

## 5.60 ArcMCCSOAP::MCC\_SOAP\_Service Class Reference

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

Inheritance diagram for ArcMCCSOAP::MCC\_SOAP\_Service:



### 5.60.1 Detailed Description

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

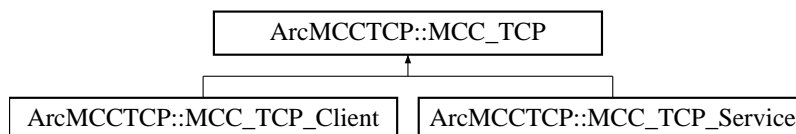
The documentation for this class was generated from the following file:

- MCCSOAP.h

## 5.61 ArcMCCTCP::MCC\_TCP Class Reference

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

Inheritance diagram for ArcMCCTCP::MCC\_TCP:



### 5.61.1 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.

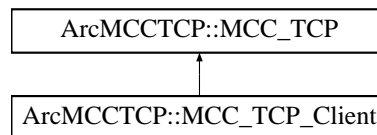
The documentation for this class was generated from the following file:

- MCCTCP.h

## 5.62 ArcMCCTCP::MCC\_TCP\_Client Class Reference

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

Inheritance diagram for ArcMCCTCP::MCC\_TCP\_Client:



### 5.62.1 Detailed Description

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

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

- MCCTCP.h

## 5.63 ArcMCCTCP::MCC\_TCP\_Service::mcc\_tcp\_exec\_t Class - Reference

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

- MCCTCP.h

## 5.64 ArcMCCTCP::MCC\_TCP\_Service::mcc\_tcp\_handle\_t Class - Reference

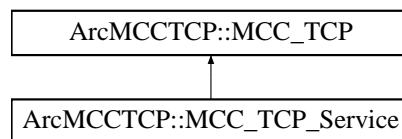
The documentation for this class was generated from the following file:

- MCCTCP.h

## 5.65 ArcMCCTCP::MCC\_TCP\_Service Class Reference

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

Inheritance diagram for ArcMCCTCP::MCC\_TCP\_Service:



### Data Structures

- class [mcc\\_tcp\\_exec\\_t](#)
- class [mcc\\_tcp\\_handle\\_t](#)

### Public Member Functions

- [MCC\\_TCP\\_Service](#) (Config \*cfg)

#### 5.65.1 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

#### 5.65.2 Constructor & Destructor Documentation

##### 5.65.2.1 ArcMCCTCP::MCC\_TCP\_Service::MCC\_TCP\_Service ( Config \* cfg )

executing function for connection thread

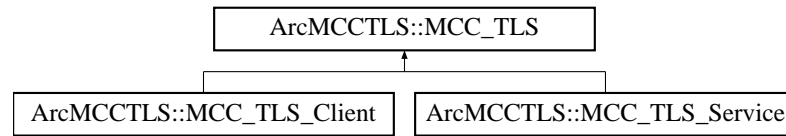
The documentation for this class was generated from the following file:

- MCCTCP.h

## 5.66 ArcMCCTLS::MCC\_TLS Class Reference

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

Inheritance diagram for ArcMCCTLS::MCC\_TLS:



### 5.66.1 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.

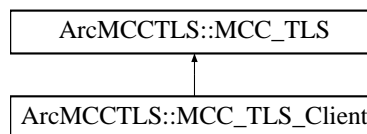
The documentation for this class was generated from the following file:

- MCCTLS.h

## 5.67 ArcMCCTLS::MCC\_TLS\_Client Class Reference

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

Inheritance diagram for ArcMCCTLS::MCC\_TLS\_Client:



### 5.67.1 Detailed Description

This class is MCC implementing TLS client.

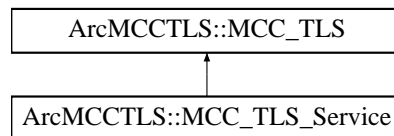
The documentation for this class was generated from the following file:

- MCCTLS.h

## 5.68 ArcMCCTLS::MCC\_TLS\_Service Class Reference

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

Inheritance diagram for ArcMCCTLS::MCC\_TLS\_Service:



### 5.68.1 Detailed Description

This MCC implements TLS server side functionality. Upon creation this object creates SSL\_CTX object and configures SSL\_CTX object with some environment information about credential. Because we cannot know the "socket" when the creation of MCC\_TLS\_Service/MCC\_TLS\_Client object (not like MCC\_TCP\_Client, which can create socket in the constructor method by using information in configuration file), we can only create "ssl" object which is binded to specified "socket", when MCC\_HTTP\_Client calls the process() method of [MCC\\_TLS\\_Client](#) object, or MCC\_TCP\_Service calls the process() method of [MCC\\_TLS\\_Service](#) object. The "ssl" object is embeded in a payload called PayloadTLSSocket.

The process() method of [MCC\\_TLS\\_Service](#) is passed payload implementing PayloadStreamInterface and the method returns empty PayloadRaw payload in "outmsg". The ssl object is created and bound to Stream payload when constructing the PayloadTLSSocket in the process() method.

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

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

- MCCTLS.h

## 5.69 ArcMCCGSI::PayloadGSISStream Class Reference

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

- PayloadGSISStream.h

## 5.70 ArcMCCHTTP::PayloadHTTP Class Reference

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

## Public Member Functions

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

## Protected Member Functions

- bool [readline](#) (std::string &line)
- bool [read](#) (char \*buf, int64\_t &size)
- bool [read\\_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\\_](#)
- int64\_t [end\\_](#)
- chunked\_t [chunked\\_](#)
- int64\_t [chunk\\_size\\_](#)
- std::multimap< std::string, std::string > [attributes\\_](#)

### 5.70.1 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.

## 5.70.2 Constructor & Destructor Documentation

5.70.2.1 `ArcMCCHTTP::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.

5.70.2.2 `ArcMCCHTTP::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.

5.70.2.3 `ArcMCCHTTP::PayloadHTTP::PayloadHTTP ( const std::string & method, const std::string & url )`

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

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

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

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

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

## 5.70.3 Member Function Documentation

5.70.3.1 `virtual const std::string& ArcMCCHTTP::PayloadHTTP::Attribute ( const std::string & name ) [virtual]`

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

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

Adds HTTP header attribute '*name*' = '*value*'

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

Returns all HTTP header attributes.

5.70.3.4 `virtual void ArcMCCHTTP::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.

5.70.3.5 `virtual bool ArcMCCHTTP::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.

5.70.3.6 `bool ArcMCCHTTP::PayloadHTTP::get_body ( void ) [protected]`

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

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

Read up to 'size' bytes from stream\_

5.70.3.8 `bool ArcMCCHTTP::PayloadHTTP::read_header ( void ) [protected]`

Read HTTP header and fill internal variables

5.70.3.9 `bool ArcMCCHTTP::PayloadHTTP::readline ( std::string & line ) [protected]`

Read from stream till

## 5.70.4 Field Documentation

5.70.4.1 `std::multimap<std::string, std::string> ArcMCCHTTP::PayloadHTTP::attributes_ [protected]`

true if conection should not be closed after response



5.70.4.2 `bool ArcMCCHTTP::PayloadHTTP::body_own_` [protected]

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

5.70.4.3 `int64_t ArcMCCHTTP::PayloadHTTP::chunk_size_` [protected]

chunked encoding parsing state

5.70.4.4 `chunked_t ArcMCCHTTP::PayloadHTTP::chunked_` [protected]

Logical end of content computed from Content-Range

5.70.4.5 `int ArcMCCHTTP::PayloadHTTP::code_` [protected]

HTTP method being used or requested

5.70.4.6 `int64_t ArcMCCHTTP::PayloadHTTP::end_` [protected]

Content-length of HTTP message

5.70.4.7 `int64_t ArcMCCHTTP::PayloadHTTP::length_` [protected]

HTTP reason being sent or supplied

5.70.4.8 `std::string ArcMCCHTTP::PayloadHTTP::method_` [protected]

minor number of HTTP version - must be 0 or 1

5.70.4.9 `PayloadRawInterface* ArcMCCHTTP::PayloadHTTP::rbody_`  
[protected]

if true stream\_ is owned by this

5.70.4.10 `std::string ArcMCCHTTP::PayloadHTTP::reason_` [protected]

HTTP code being sent or supplied

5.70.4.11 `PayloadStreamInterface* ArcMCCHTTP::PayloadHTTP::sbody_`  
[protected]

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

#### 5.70.4.12 `PayloadStreamInterface*` `ArcMCCHTTP::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_`.

#### 5.70.4.13 `bool` `ArcMCCHTTP::PayloadHTTP::stream_own_` `[protected]`

stream used to communicate to outside

#### 5.70.4.14 `std::string` `ArcMCCHTTP::PayloadHTTP::uri_` `[protected]`

if true `body_` is owned by this

#### 5.70.4.15 `int` `ArcMCCHTTP::PayloadHTTP::version_major_` `[protected]`

URI being contacted

#### 5.70.4.16 `int` `ArcMCCHTTP::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`

## 5.71 `ArcMCCTCP::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)

#### 5.71.1 Detailed Description

This class extends `PayloadStream` with TCP socket specific features

### 5.71.2 Constructor & Destructor Documentation

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

Constructor - connects to TCP server at specified hostname:port

5.71.2.2 `ArcMCCTCP::PayloadTCPSocket::PayloadTCPSocket ( const std::string & endpoint, int timeout, Logger & logger )`

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

5.71.2.3 `ArcMCCTCP::PayloadTCPSocket::PayloadTCPSocket ( int s, int timeout, Logger & logger ) [inline]`

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

5.71.2.4 `ArcMCCTCP::PayloadTCPSocket::PayloadTCPSocket ( PayloadTCPSocket & s ) [inline]`

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

5.71.2.5 `ArcMCCTCP::PayloadTCPSocket::PayloadTCPSocket ( PayloadTCPSocket & s, Logger & logger ) [inline]`

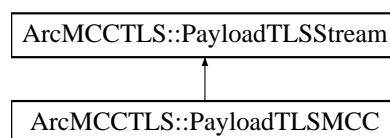
Copy constructor - inherits handle of copied object. Handle is NOT closed in destructor.

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

- PayloadTCPSocket.h

## 5.72 ArcMCCTLS::PayloadTLMCC Class Reference

Inheritance diagram for ArcMCCTLS::PayloadTLMCC:



## Public Member Functions

- [PayloadTLMCC](#) (MCCInterface \*mcc, const [ConfigTLMCC](#) &cfg, Logger &logger)
- [PayloadTLMCC](#) (PayloadStreamInterface \*stream, const [ConfigTLMCC](#) &cfg, Logger &logger)
- [PayloadTLMCC](#) ([PayloadTLMCC](#) &stream)

## 5.72.1 Constructor & Destructor Documentation

### 5.72.1.1 ArcMCCTLS::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.

### 5.72.1.2 ArcMCCTLS::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.

### 5.72.1.3 ArcMCCTLS::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.

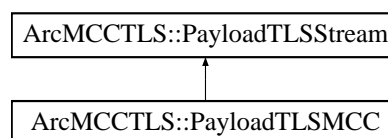
The documentation for this class was generated from the following file:

- [PayloadTLMCC.h](#)

## 5.73 ArcMCCTLS::PayloadTLSStream Class Reference

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

Inheritance diagram for ArcMCCTLS::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\\_](#)

### 5.73.1 Detailed Description

Implemetation of PayloadStreamInterface for SSL handle.

### 5.73.2 Constructor & Destructor Documentation

**5.73.2.1** ArcMCCTLS::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.

**5.73.2.2** virtual ArcMCCTLS::PayloadTLSStream::~~PayloadTLSStream ( void )  
[virtual]

Destructor.

### 5.73.3 Member Function Documentation

**5.73.3.1** X509\* ArcMCCTLS::PayloadTLSStream::GetCert ( void )

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

**5.73.3.2** X509\* ArcMCCTLS::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.

### 5.73.3.3 ArcMCCTLS::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.

## 5.73.4 Field Documentation

### 5.73.4.1 SSL\* ArcMCCTLS::PayloadTLSStream::ssl\_ [protected]

Timeout for read/write operations

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

- PayloadTLSStream.h

## 5.74 ArcSec::ArcAuthZ::PDPDesc Class Reference

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

- ArcAuthZ.h

## 5.75 ArcSec::PDPServiceInvoker Class Reference

```
#include <PDPServiceInvoker.h>
```

### 5.75.1 Detailed Description

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

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

- PDPServiceInvoker.h

## 5.76 ArcSec::SAML2SSO\_AssertionConsumerSH Class Reference

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

### 5.76.1 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

## 5.77 ArcSec::SAMLTokenSH Class Reference

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

### 5.77.1 Detailed Description

Adds WS-Security SAML Token into SOAP Header.

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

- SAMLTokenSH.h

## 5.78 ArcSec::SimpleListPDP Class Reference

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

### 5.78.1 Detailed Description

Tests X509 subject against list of subjects in file.

This class implements PDP interface. It's isPermitted() method compares X590 subject of requestor obtained from TLS layer (TLS:PEERDN) to list of subjects (one per line) in external file. Locations 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

## 5.79 ArcSHCLegacy::SimpleMap Class Reference

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

- simplemap.h

## 5.80 ArcSHCLegacy::AuthUser::source\_t Struct Reference

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

- auth.h

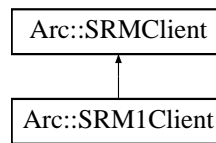
## 5.81 ArcSHCLegacy::UnixMap::source\_t Struct Reference

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

- unixmap.h

## 5.82 Arc::SRM1Client Class Reference

Inheritance diagram for Arc::SRM1Client:



### Public Member Functions

- SRMReturnCode [ping](#) (std::string &, bool=true)
- SRMReturnCode [getSpaceTokens](#) (std::list< std::string > &, const std::string &= "")
- SRMReturnCode [getRequestTokens](#) (std::list< std::string > &, const std::string &= "")
- SRMReturnCode [requestBringOnline](#) (SRMClientRequest &)
- SRMReturnCode [requestBringOnlineStatus](#) (SRMClientRequest &)
- SRMReturnCode [mkDir](#) (SRMClientRequest &)
- SRMReturnCode [checkPermissions](#) (SRMClientRequest &)
- SRMReturnCode [getURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [getURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [putURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [putURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- SRMReturnCode [releaseGet](#) (SRMClientRequest &req)
- SRMReturnCode [releasePut](#) (SRMClientRequest &req)
- SRMReturnCode [release](#) (SRMClientRequest &req)
- SRMReturnCode [abort](#) (SRMClientRequest &req)
- SRMReturnCode [info](#) (SRMClientRequest &req, std::map< std::string, std::list< struct [SRMFileMetaData](#) > > &metadata)
- SRMReturnCode [info](#) (SRMClientRequest &req, std::list< struct [SRMFileMetaData](#) > &metadata)
- SRMReturnCode [remove](#) (SRMClientRequest &req)
- SRMReturnCode [copy](#) (SRMClientRequest &req, const std::string &source)



### 5.82.1 Member Function Documentation

**5.82.1.1** `SRMReturnCode Arc::SRM1Client::abort ( SRMClientRequest & req )`  
[virtual]

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

#### Parameters

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

#### Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

**5.82.1.2** `SRMReturnCode Arc::SRM1Client::checkPermissions ( SRMClientRequest & req )`  
[inline, virtual]

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

#### Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

Copy a file between two SRM storages.

#### Parameters

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

#### Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

#### 5.82.1.4 SRMReturnCode Arc::SRM1Client::getRequestTokens ( std::list< std::string > & tokens, const std::string & description = " " ) [inline, virtual]

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

##### Parameters

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

##### Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

#### 5.82.1.5 SRMReturnCode Arc::SRM1Client::getSpaceTokens ( std::list< std::string > & tokens, const std::string & description = " " ) [inline, virtual]

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

##### Parameters

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

##### Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

##### Parameters

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

**5.82.1.8** SRMReturnCode Arc::SRM1Client::info ( SRMClientRequest & req, std::map< std::string, std::list< struct SRMFileMetaData > > & metadata ) [virtual]

Returns information on a file or files (v2.2 and higher) stored in SRM, such as file size, checksum and estimated access latency. If a directory or directories is listed with recursion >= 1 then the list mapped to each SURL in metadata will contain the content of the directory or directories.

**Parameters**

<i>req</i>	The request object
<i>metadata</i>	A map mapping each SURL in the request to a list of structs filled with file information. If a SURL is missing from the map it means there was some problem accessing it.

**Returns**

SRMReturnCode specifying outcome of operation

**See also**

[SRMFileMetaData](#)

Implements [Arc::SRMClient](#).

**5.82.1.9** `SRMReturnCode Arc::SRM1Client::info ( SRMClientRequest & req, std::list< struct SRMFileMetaData > & metadata ) [virtual]`

Returns information on a file stored in an SRM, such as file size, checksum and estimated access latency. If a directory is listed with recursion  $\geq 1$  then the list in metadata will contain the content of the directory.

#### Parameters

<i>req</i>	The request object
<i>metadata</i>	A list of structs filled with file information.

#### Returns

SRMReturnCode specifying outcome of operation

#### See also

[SRMFileMetaData](#)

Implements [Arc::SRMClient](#).

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

Make required directories for the SURL in the request

#### Parameters

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

#### Returns

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

**5.82.1.11** `SRMReturnCode Arc::SRM1Client::ping ( std::string & version, bool report_error = true ) [inline, virtual]`

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

#### Parameters

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

Should be called after a successful copy from SRM storage.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

Should be called after a successful copy to SRM storage.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

Delete a file physically from storage and the SRM namespace.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

Implements [Arc::SRMClient](#).

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

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

## Parameters

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

## Returns

SRMReturnCode specifying outcome of operation

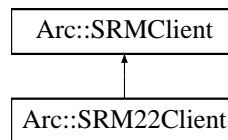
Implements [Arc::SRMClient](#).

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

- SRM1Client.h

### 5.83 Arc::SRM22Client Class Reference

Inheritance diagram for Arc::SRM22Client:



#### Public Member Functions

- [SRM22Client](#) (const UserConfig &usercfg, const [SRMURL](#) &url)
- [~SRM22Client](#) ()
- SRMReturnCode [ping](#) (std::string &version, bool report\_error=true)
- SRMReturnCode [getSpaceTokens](#) (std::list< std::string > &tokens, const std::string &description="")
- SRMReturnCode [getRequestTokens](#) (std::list< std::string > &tokens, const std::string &description="")
- SRMReturnCode [getURLs](#) ([SRMClientRequest](#) &req, std::list< std::string > &urls)
- SRMReturnCode [getURLsStatus](#) ([SRMClientRequest](#) &req, std::list< std::string > &urls)
- SRMReturnCode [putURLs](#) ([SRMClientRequest](#) &req, std::list< std::string > &urls)
- SRMReturnCode [putURLsStatus](#) ([SRMClientRequest](#) &req, std::list< std::string > &urls)
- SRMReturnCode [requestBringOnline](#) ([SRMClientRequest](#) &req)
- SRMReturnCode [requestBringOnlineStatus](#) ([SRMClientRequest](#) &req)
- SRMReturnCode [info](#) ([SRMClientRequest](#) &req, std::map< std::string, std::list< struct [SRMFileMetaData](#) > > &metadata)
- SRMReturnCode [info](#) ([SRMClientRequest](#) &req, std::list< struct [SRMFileMetaData](#) > &metadata)



- SRMReturnCode [releaseGet](#) (SRMClientRequest &req)
- SRMReturnCode [releasePut](#) (SRMClientRequest &req)
- SRMReturnCode [release](#) (SRMClientRequest &)
- SRMReturnCode [abort](#) (SRMClientRequest &req)
- SRMReturnCode [remove](#) (SRMClientRequest &req)
- SRMReturnCode [copy](#) (SRMClientRequest &req, const std::string &source)
- SRMReturnCode [mkDir](#) (SRMClientRequest &req)
- SRMReturnCode [checkPermissions](#) (SRMClientRequest &req)

### 5.83.1 Constructor & Destructor Documentation

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

Constructor

5.83.1.2 **Arc::SRM22Client::~SRM22Client** ( )

Destructor

### 5.83.2 Member Function Documentation

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

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

Implements [Arc::SRMClient](#).

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

Call srmCheckPermission

Implements [Arc::SRMClient](#).

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

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

Implements [Arc::SRMClient](#).

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

Use `srmGetRequestTokens` to return a list of spaces available

Implements [Arc::SRMClient](#).

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

Use `srmGetSpaceTokens` to return a list of spaces available

Implements [Arc::SRMClient](#).

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

Get a list of TURLs for the given SURL. Uses `srmPrepareToGet` and waits until file is ready (online and pinned) if the request is synchronous. If not it returns after making the request. Although a list is returned, SRMv2.2 only returns one TURL per SURL.

Implements [Arc::SRMClient](#).

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

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

Implements [Arc::SRMClient](#).

**5.83.2.8** `SRMReturnCode Arc::SRM22Client::info ( SRMClientRequest & req, std::map< std::string, std::list< struct SRMFileMetaData > > & metadata ) [virtual]`

Use `srmLs` to get info on the given SURLs. Info on each file or content of directory is put in a list of metadata structs.

Implements [Arc::SRMClient](#).

**5.83.2.9** `SRMReturnCode Arc::SRM22Client::info ( SRMClientRequest & req, std::list< struct SRMFileMetaData > & metadata ) [virtual]`

Use `srmLs` to get info on the given SURL. Info on each file or content of directory is put in a list of metadata structs

Implements [Arc::SRMClient](#).

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

Call srmMkDir

Implements [Arc::SRMClient](#).

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

Get the server version from srmPing

Implements [Arc::SRMClient](#).

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

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

Implements [Arc::SRMClient](#).

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

Uses srmStatusOfPutRequest to query the status of the given request.

Implements [Arc::SRMClient](#).

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

Not used in this version of SRM

Implements [Arc::SRMClient](#).

**5.83.2.15** `SRMReturnCode Arc::SRM22Client::releaseGet ( SRMClientRequest & req )`  
[virtual]

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

Implements [Arc::SRMClient](#).

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

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

Implements [Arc::SRMClient](#).

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

Delete by srmRm or srmRmDir

Implements [Arc::SRMClient](#).

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

Call srmBringOnline with the URLs specified in req.

Implements [Arc::SRMClient](#).

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

Call srmStatusOfBringOnlineRequest and update req with any changes.

Implements [Arc::SRMClient](#).

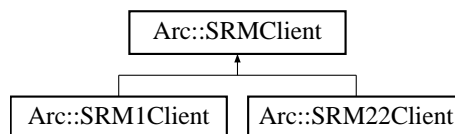
The documentation for this class was generated from the following file:

- SRM22Client.h

## 5.84 Arc::SRMClient Class Reference

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

Inheritance diagram for Arc::SRMClient:



### Public Member Functions

- virtual [~SRMClient](#) ()

- std::string [getVersion](#) () const
- virtual SRMReturnCode [ping](#) (std::string &version, bool report\_error=true)=0
- virtual SRMReturnCode [getSpaceTokens](#) (std::list< std::string > &tokens, const std::string &description="")=0
- virtual SRMReturnCode [getRequestTokens](#) (std::list< std::string > &tokens, const std::string &description="")=0
- virtual SRMReturnCode [getURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode [getURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode [requestBringOnline](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [requestBringOnlineStatus](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [putURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode [putURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)=0
- virtual SRMReturnCode [releaseGet](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [releasePut](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [release](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [abort](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [info](#) (SRMClientRequest &req, std::map< std::string, std::list< struct [SRMFileMetaData](#) > > &metadata)=0
- virtual SRMReturnCode [info](#) (SRMClientRequest &req, std::list< struct [SRMFileMetaData](#) > &metadata)=0
- virtual SRMReturnCode [remove](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [copy](#) (SRMClientRequest &req, const std::string &source)=0
- virtual SRMReturnCode [mkDir](#) (SRMClientRequest &req)=0
- virtual SRMReturnCode [checkPermissions](#) (SRMClientRequest &req)=0

### Static Public Member Functions

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

### Protected Member Functions

- [SRMClient](#) (const UserConfig &usercfg, const [SRMURL](#) &url)
- SRMReturnCode [process](#) (const std::string &action, PayloadSOAP \*request, - PayloadSOAP \*\*response)

### Protected Attributes

- std::string [service\\_endpoint](#)
- MCCCConfig [cfg](#)
- ClientSOAP \* [client](#)

- NS [ns](#)
- SRMImplementation [implementation](#)
- time\_t [user\\_timeout](#)
- std::string [version](#)

### Static Protected Attributes

- static Logger [logger](#)

### 5.84.1 Detailed Description

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

### 5.84.2 Constructor & Destructor Documentation

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

Destructor

### 5.84.3 Member Function Documentation

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

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

#### Parameters

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

#### Returns

SRMReturnCode specifying outcome of operation

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

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

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

Copy a file between two SRM storages.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

Create an [SRMClient](#) instance. The instance will be a SRM v2.2 client unless another version is explicitly given in the url.

**Parameters**

<i>usercfg</i>	The user configuration.
<i>url</i>	A SURL. A client connects to the service host derived from this SURL. All operations with a client instance must use SURLs with the same host as this one.
<i>timedout</i>	Whether the connection timed out

**Returns**

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

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

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

## Parameters

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

## Returns

SRMReturnCode specifying outcome of operation

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

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

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

## Parameters

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

## Returns

SRMReturnCode specifying outcome of operation

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

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

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

## Parameters

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

## Returns

SRMReturnCode specifying outcome of operation

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



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

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

#### Parameters

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

#### Returns

SRMReturnCode specifying outcome of operation

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

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

Returns the version of the SRM protocol used by this instance

References version.

**5.84.3.10** `virtual SRMReturnCode Arc::SRMClient::info ( SRMClientRequest & req, std::map< std::string, std::list< struct SRMFileMetaData > > & metadata ) [pure virtual]`

Returns information on a file or files (v2.2 and higher) stored in SRM, such as file size, checksum and estimated access latency. If a directory or directories is listed with recursion  $\geq 1$  then the list mapped to each SURL in metadata will contain the content of the directory or directories.

#### Parameters

<i>req</i>	The request object
<i>metadata</i>	A map mapping each SURL in the request to a list of structs filled with file information. If a SURL is missing from the map it means there was some problem accessing it.

#### Returns

SRMReturnCode specifying outcome of operation

#### See also

[SRMFileMetaData](#)

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

5.84.3.11 `virtual SRMReturnCode Arc::SRMClient::info ( SRMClientRequest & req,  
std::list< struct SRMFileMetaData > & metadata ) [pure virtual]`

Returns information on a file stored in an SRM, such as file size, checksum and estimated access latency. If a directory is listed with recursion  $\geq 1$  then the list in metadata will contain the content of the directory.

#### Parameters

<i>req</i>	The request object
<i>metadata</i>	A list of structs filled with file information.

#### Returns

SRMReturnCode specifying outcome of operation

#### See also

[SRMFileMetaData](#)

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

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

Make required directories for the SURL in the request

#### Parameters

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

#### Returns

SRMReturnCode specifying outcome of operation

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

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

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

#### Parameters

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

Should be called after a successful copy from SRM storage.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

Should be called after a successful copy to SRM storage.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

Delete a file physically from storage and the SRM namespace.

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

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

**Parameters**

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

**Returns**

SRMReturnCode specifying outcome of operation

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

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

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

## Parameters

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

## Returns

SRMReturnCode specifying outcome of operation

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

#### 5.84.4 Field Documentation

##### 5.84.4.1 `std::string Arc::SRMClient::service_endpoint` [protected]

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

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

- `SRMClient.h`

### 5.85 `Arc::SRMClientRequest` Class Reference

```
#include <SRMClientRequest.h>
```

#### Public Member Functions

- [SRMClientRequest](#) (const std::list< std::string > &urls) throw (SRMInvalidRequestException)
- [SRMClientRequest](#) (const std::string &url="", const std::string &id="") throw (SRMInvalidRequestException)
- std::string [surl](#) () const
- int [waiting\\_time](#) () const
- void [finished\\_success](#) ()
- void [finished\\_partial\\_success](#) ()
- void [finished\\_error](#) ()
- void [finished\\_abort](#) ()
- void [wait](#) (int t=0)
- void [cancelled](#) ()
- SRMRequestStatus [status](#) () const

#### 5.85.1 Detailed Description

Class to represent a SRM request.

It may be used for multiple operations, for example calling `getURLs()` sets the request token in the request object (for a v2.2 client) and then same object is passed to `release-Get()`.

### 5.85.2 Constructor & Destructor Documentation

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

Creates a request object with multiple SURLS.

The URLs here are in the form srm://srm.host.org/path/to/file

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

Creates a request object with a single SURL.

The URL here is in the form srm://srm.host.org/path/to/file

### 5.85.3 Member Function Documentation

**5.85.3.1** `int Arc::SRMClientRequest::waiting_time ( ) const [inline]`

Get waiting time. A waiting time of zero means no estimate was given by the remote service.

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

- SRMClientRequest.h

## 5.86 SRMFileInfo Class Reference

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

### 5.86.1 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

## 5.87 Arc::SRMFileMetaData Struct Reference

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

### 5.87.1 Detailed Description

SRM-related file metadata.

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

- [SRMClient.h](#)

## 5.88 SRMInfo Class Reference

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

### 5.88.1 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](#)

## 5.89 Arc::SRMInvalidRequestException Class Reference

```
#include <SRMClientRequest.h>
```

### 5.89.1 Detailed Description

General exception to represent a bad SRM request.

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

- [SRMClientRequest.h](#)

## 5.90 SRMURL Class Reference

### Public Member Functions

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



- std::string [ShortURL](#) (void) const
- std::string [FullURL](#) (void) const
- bool [PortDefined](#) ()

## 5.90.1 Constructor & Destructor Documentation

### 5.90.1.1 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

## 5.90.2 Member Function Documentation

### 5.90.2.1 std::string SRMURL::BaseURL ( void ) const

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

### 5.90.2.2 std::string SRMURL::ContactURL ( void ) const

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

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

eg /srm/managerv2

### 5.90.2.4 std::string SRMURL::FileName ( void ) const [inline]

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

### 5.90.2.5 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

### 5.90.2.6 bool SRMURL::PortDefined ( ) [inline]

Was the port number given in the constructor?

### 5.90.2.7 void SRMURL::SetSRMVersion ( const std::string & version )

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

#### 5.90.2.8 `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

### 5.91 `ArcSHCLegacy::UnixMap::unix_user_t` Class Reference

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

- unixmap.h

### 5.92 `ArcSHCLegacy::UnixMap` Class Reference

#### Data Structures

- struct [source\\_t](#)
- class [unix\\_user\\_t](#)

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

- unixmap.h

### 5.93 `ArcSec::UsernameTokenSH` Class Reference

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

#### 5.93.1 Detailed Description

Adds WS-Security Username Token into SOAP Header.

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

- UsernameTokenSH.h

### 5.94 `ArcSHCLegacy::voms` Struct Reference

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

## Data Fields

- std::string [server](#)
- std::string [voname](#)
- std::vector< [voms\\_attrs](#) > [attrs](#)

### 5.94.1 Detailed Description

VOMS data

### 5.94.2 Field Documentation

#### 5.94.2.1 std::vector<voms\_attrs> ArcSHCLegacy::voms::attrs

User's characteristics

#### 5.94.2.2 std::string ArcSHCLegacy::voms::server

The VOMS server DN, as from its certificate

#### 5.94.2.3 std::string ArcSHCLegacy::voms::voname

The name of the VO to which the VOMS belongs

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

- [auth.h](#)

## 5.95 ArcSHCLegacy::voms\_attrs Struct Reference

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

## Data Fields

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

### 5.95.1 Detailed Description

VOMS attributes

### 5.95.2 Field Documentation

#### 5.95.2.1 `std::string ArcSHCLegacy::voms_attrs::cap`

user's capability

#### 5.95.2.2 `std::string ArcSHCLegacy::voms_attrs::group`

user's group

#### 5.95.2.3 `std::string ArcSHCLegacy::voms_attrs::role`

user's role

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

- `auth.h`

## 5.96 `ArcSec::X509TokenSH` Class Reference

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

### 5.96.1 Detailed Description

Adds WS-Security X509 Token into SOAP Header.

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

- `X509TokenSH.h`

## 5.97 `ArcSec::XACMLAlgFactory` Class Reference

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

### Public Member Functions

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

### 5.97.1 Detailed Description

Algorithm factory class for XACML.

### 5.97.2 Member Function Documentation

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

return a Alg object according to the "CombiningAlg" attribute in the <Policy> node; The [XACMLAlgFactory](#) itself will release the Alg objects

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

- XACMLAlgFactory.h

## 5.98 ArcSec::XACMLApply Class Reference

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

- XACMLApply.h

## 5.99 ArcSec::XACMLAttributeFactory Class Reference

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

### Public Member Functions

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

### 5.99.1 Detailed Description

Attribute factory class for XACML specified attributes.

### 5.99.2 Member Function Documentation

5.99.2.1 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

## 5.100 ArcSec::XACMLAttributeProxy Class Reference

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

### Public Member Functions

- virtual AttributeValue \* [getAttribute](#) (const Arc::XMLNode &node)

#### 5.100.1 Detailed Description

XACML specific AttributeProxy class.

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

- XACMLAttributeProxy.h

## 5.101 ArcSec::XACMLCondition Class Reference

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

### Public Member Functions

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

#### 5.101.1 Detailed Description

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

#### 5.101.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLCondition.h

## 5.102 ArcSec::XACMLEvaluationCtx Class Reference

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

## Public Member Functions

- [XACMLEvaluationCtx](#) (Request \*request)

### 5.102.1 Detailed Description

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

### 5.102.2 Constructor & Destructor Documentation

#### 5.102.2.1 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

## 5.103 ArcSec::XACMLEvaluator Class Reference

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

## Public Member Functions

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

### 5.103.1 Detailed Description

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

### 5.103.2 Member Function Documentation

#### 5.103.2.1 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

## 5.104 ArcSec::XACMLFnFactory Class Reference

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

### Public Member Functions

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

#### 5.104.1 Detailed Description

Function factory class for XACML specified attributes.

#### 5.104.2 Member Function Documentation

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

return a Function object according to the "Function" attribute in the XML node; The [XACMLFnFactory](#) itself will release the Function objects

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

- XACMLFnFactory.h

## 5.105 ArcSec::XACMLPDP Class Reference

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

#### 5.105.1 Detailed Description

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

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

- XACMLPDP.h

## 5.106 ArcSec::XACMLPolicy Class Reference

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



## Public Member Functions

- [XACMLPolicy](#) (void)
- [XACMLPolicy](#) (const Arc::XMLNode node)
- [XACMLPolicy](#) (const Arc::XMLNode node, EvaluatorContext \*ctx)
- virtual void [make\\_policy](#) ()

### 5.106.1 Detailed Description

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

### 5.106.2 Constructor & Destructor Documentation

#### 5.106.2.1 ArcSec::XACMLPolicy::XACMLPolicy ( void )

Constructor

#### 5.106.2.2 ArcSec::XACMLPolicy::XACMLPolicy ( const Arc::XMLNode *node* )

Constructor

#### 5.106.2.3 ArcSec::XACMLPolicy::XACMLPolicy ( const Arc::XMLNode *node*, EvaluatorContext \* *ctx* )

Constructor -

### 5.106.3 Member Function Documentation

#### 5.106.3.1 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

## 5.107 ArcSec::XACMLRequest Class Reference

### Public Member Functions

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

### 5.107.1 Member Function Documentation

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

Get the name of corresponding evaluator

5.107.1.2 `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

## 5.108 ArcSec::XACMLRule Class Reference

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

### 5.108.1 Detailed Description

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

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

- XACMLRule.h

## 5.109 ArcSec::XACMLTarget Class Reference

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

### Public Member Functions

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

### 5.109.1 Detailed Description

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

## 5.109.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLTarget.h

## 5.110 ArcSec::XACMLTargetMatch Class Reference

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

- XACMLTarget.h

## 5.111 ArcSec::XACMLTargetMatchGroup Class Reference

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

- XACMLTarget.h

## 5.112 ArcSec::XACMLTargetSection Class Reference

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

- XACMLTarget.h