

Hosting Environment (Daemon) Chain Components

Generated by Doxygen 1.6.1

Sat Feb 2 01:01:22 2013

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Data Structure Index	3
2.1	Class Hierarchy	3
3	Data Structure Index	7
3.1	Data Structures	7
4	Namespace Documentation	11
4.1	ArcSec Namespace Reference	11
4.1.1	Detailed Description	13
4.1.2	Typedef Documentation	13
4.1.2.1	AndList	13
4.1.2.2	Match	14
5	Data Structure Documentation	15
5.1	ArcSec::AllowPDP Class Reference	15
5.1.1	Detailed Description	15
5.2	ArcSec::ArcAlgFactory Class Reference	16
5.2.1	Detailed Description	16
5.2.2	Member Function Documentation	16
5.2.2.1	createAlg	16
5.3	ArcSec::ArcAttributeFactory Class Reference	17
5.3.1	Detailed Description	17
5.3.2	Member Function Documentation	17
5.3.2.1	createValue	17
5.4	ArcSec::ArcAttributeProxy< TheAttribute > Class Template Reference	18
5.4.1	Detailed Description	18
5.5	ArcSec::ArcAuthZ Class Reference	19

5.5.1	Detailed Description	19
5.5.2	Member Function Documentation	19
5.5.2.1	Handle	19
5.5.2.2	MakePDPs	19
5.6	ArcSec::ArcEvaluationCtx Class Reference	20
5.6.1	Detailed Description	20
5.6.2	Constructor & Destructor Documentation	20
5.6.2.1	ArcEvaluationCtx	20
5.6.3	Member Function Documentation	20
5.6.3.1	split	20
5.7	ArcSec::ArcEvaluator Class Reference	21
5.7.1	Detailed Description	21
5.7.2	Member Function Documentation	21
5.7.2.1	evaluate	21
5.8	ArcSec::ArcFnFactory Class Reference	22
5.8.1	Detailed Description	22
5.8.2	Member Function Documentation	22
5.8.2.1	createFn	22
5.9	ArcSec::ArcPDP Class Reference	23
5.9.1	Detailed Description	23
5.10	ArcSec::ArcPolicy Class Reference	24
5.10.1	Detailed Description	24
5.10.2	Constructor & Destructor Documentation	24
5.10.2.1	ArcPolicy	24
5.10.2.2	ArcPolicy	24
5.10.2.3	ArcPolicy	24
5.10.3	Member Function Documentation	24
5.10.3.1	make_policy	24
5.11	ArcSec::ArcRequest Class Reference	25
5.12	ArcSec::ArcRequestItem Class Reference	26
5.12.1	Detailed Description	26
5.13	ArcSec::ArcRequestTuple Class Reference	27
5.13.1	Detailed Description	27
5.14	ArcSec::ArcRule Class Reference	28
5.14.1	Detailed Description	28
5.15	ArcSec::AttributeDesignator Class Reference	29

5.16	ArcSec::AttributeSelector Class Reference	30
5.17	ArcSHCLegacy::AuthUser Class Reference	31
5.18	ArcSHCLegacy::AuthVO Class Reference	32
5.19	ArcSHCLegacy::ConfigParser Class Reference	33
5.20	ArcMCCTLS::ConfigTlsmcc Class Reference	34
5.21	ArcDMCARC::DataPointARC Class Reference	35
5.21.1	Detailed Description	35
5.22	ArcDMCFile::DataPointFile Class Reference	36
5.22.1	Detailed Description	36
5.23	ArcDMCGFAL::DataPointGFAL Class Reference	37
5.23.1	Detailed Description	37
5.24	ArcDMCGridFTP::DataPointGridFTP Class Reference	38
5.24.1	Detailed Description	38
5.25	ArcDMCHTTP::DataPointHTTP Class Reference	39
5.25.1	Detailed Description	39
5.26	ArcDMCLDAP::DataPointLDAP Class Reference	40
5.26.1	Detailed Description	40
5.27	ArcDMCLFC::DataPointLFC Class Reference	41
5.27.1	Detailed Description	41
5.28	ArcDMCMock::DataPointMock Class Reference	42
5.28.1	Detailed Description	42
5.29	ArcDMCSR::DataPointSRM Class Reference	43
5.29.1	Detailed Description	43
5.30	ArcDMCXrootd::DataPointXrootd Class Reference	44
5.30.1	Detailed Description	44
5.31	ArcMCCTLSSec::DelegationCollector Class Reference	45
5.32	ArcMCCTLSSec::DelegationMultiSecAttr Class Reference	46
5.33	ArcSec::DelegationPDP Class Reference	47
5.33.1	Detailed Description	47
5.34	ArcMCCTLSSec::DelegationSecAttr Class Reference	48
5.35	ArcSec::DelegationSH Class Reference	49
5.36	ArcSec::DenyPDP Class Reference	50
5.36.1	Detailed Description	50
5.37	ArcSec::GACLEvaluator Class Reference	51
5.37.1	Member Function Documentation	51
5.37.1.1	evaluate	51

5.38	ArcSec::GACLPDP Class Reference	52
5.39	ArcSec::GACLPolicy Class Reference	53
5.40	ArcSec::GACLRequest Class Reference	54
5.41	ArcDMCGFAL::GFALTransfer3rdParty Class Reference	55
5.41.1	Detailed Description	55
5.42	ArcDMCGFAL::GFALUtils Class Reference	56
5.42.1	Detailed Description	56
5.43	ArcDMCLDAP::LDAPQuery Class Reference	57
5.43.1	Detailed Description	57
5.43.2	Constructor & Destructor Documentation	57
5.43.2.1	LDAPQuery	57
5.43.2.2	~LDAPQuery	57
5.43.3	Member Function Documentation	57
5.43.3.1	Query	57
5.43.3.2	Result	57
5.44	ArcSHCLegacy::LegacyMap Class Reference	59
5.45	ArcSHCLegacy::LegacyPDP Class Reference	60
5.46	ArcSHCLegacy::LegacySecAttr Class Reference	61
5.47	ArcSHCLegacy::LegacySecHandler Class Reference	62
5.48	ArcDMCGridFTP::Lister Class Reference	63
5.49	ArcMCCGSI::MCC_GSI_Client Class Reference	64
5.50	ArcMCCGSI::MCC_GSI_Service Class Reference	65
5.51	ArcMCCHTTP::MCC_HTTP Class Reference	66
5.51.1	Detailed Description	66
5.52	ArcMCCHTTP::MCC_HTTP_Client Class Reference	67
5.52.1	Detailed Description	67
5.53	ArcMCCHTTP::MCC_HTTP_Service Class Reference	68
5.53.1	Detailed Description	68
5.54	ArcMCCMsgValidator::MCC_MsgValidator Class Reference	69
5.55	ArcMCCMsgValidator::MCC_MsgValidator_Service Class Reference	70
5.56	ArcMCCSOAP::MCC_SOAP Class Reference	71
5.56.1	Detailed Description	71
5.57	ArcMCCSOAP::MCC_SOAP_Client Class Reference	72
5.58	ArcMCCSOAP::MCC_SOAP_Service Class Reference	73
5.58.1	Detailed Description	73
5.59	ArcMCCTCP::MCC_TCP Class Reference	74

5.59.1 Detailed Description	74
5.60 ArcMCCTCP::MCC_TCP_Client Class Reference	75
5.60.1 Detailed Description	75
5.61 ArcMCCTCP::MCC_TCP_Service Class Reference	76
5.61.1 Detailed Description	76
5.61.2 Constructor & Destructor Documentation	76
5.61.2.1 MCC_TCP_Service	76
5.62 ArcMCCTLS::MCC_TLS Class Reference	77
5.62.1 Detailed Description	77
5.63 ArcMCCTLS::MCC_TLS_Client Class Reference	78
5.63.1 Detailed Description	78
5.64 ArcMCCTLS::MCC_TLS_Service Class Reference	79
5.64.1 Detailed Description	79
5.65 ArcMCCGSI::PayloadGSISStream Class Reference	80
5.66 ArcMCCHTTP::PayloadHTTP Class Reference	81
5.66.1 Detailed Description	81
5.66.2 Constructor & Destructor Documentation	81
5.66.2.1 PayloadHTTP	81
5.66.2.2 PayloadHTTP	82
5.66.2.3 PayloadHTTP	82
5.66.3 Member Function Documentation	82
5.66.3.1 Attribute	82
5.66.3.2 Attributes	82
5.66.3.3 GetError	82
5.66.4 Field Documentation	82
5.66.4.1 attributes_	82
5.66.4.2 code_	82
5.66.4.3 end_	82
5.66.4.4 keep_alive_	82
5.66.4.5 length_	82
5.66.4.6 method_	83
5.66.4.7 offset_	83
5.66.4.8 reason_	83
5.66.4.9 size_	83
5.66.4.10 uri_	83
5.66.4.11 version_major_	83

5.66.4.12	version_minor_	83
5.67	ArcMCCHTTP::PayloadHTTPIn Class Reference	84
5.67.1	Constructor & Destructor Documentation	84
5.67.1.1	PayloadHTTPIn	84
5.67.2	Member Function Documentation	84
5.67.2.1	get_body	84
5.67.2.2	read	85
5.67.2.3	read_header	85
5.67.2.4	readline	85
5.67.3	Field Documentation	85
5.67.3.1	body_	85
5.67.3.2	chunk_size_	85
5.67.3.3	fetched_	85
5.67.3.4	stream_offset_	85
5.67.3.5	stream_own_	85
5.67.3.6	tbuf_	85
5.67.3.7	tbuflen_	85
5.68	ArcMCCHTTP::PayloadHTTPOut Class Reference	86
5.68.1	Constructor & Destructor Documentation	86
5.68.1.1	PayloadHTTPOut	86
5.68.1.2	PayloadHTTPOut	86
5.68.2	Member Function Documentation	87
5.68.2.1	Attribute	87
5.68.2.2	Flush	87
5.68.2.3	make_header	87
5.68.3	Field Documentation	87
5.68.3.1	header_	87
5.68.3.2	rbody_	87
5.68.3.3	sbody_	87
5.68.3.4	sbody_size_	87
5.68.3.5	stream_offset_	87
5.68.3.6	to_stream_	87
5.68.3.7	use_chunked_transfer_	87
5.69	ArcMCCHTTP::PayloadHTTPOutRaw Class Reference	89
5.69.1	Member Function Documentation	89
5.69.1.1	Body	89

5.70	ArcMCCHTTP::PayloadHTTPOutStream Class Reference	90
5.70.1	Member Function Documentation	90
5.70.1.1	Body	90
5.71	ArcMCCTCP::PayloadTCPSocket Class Reference	91
5.71.1	Detailed Description	91
5.71.2	Constructor & Destructor Documentation	91
5.71.2.1	PayloadTCPSocket	91
5.71.2.2	PayloadTCPSocket	91
5.71.2.3	PayloadTCPSocket	91
5.71.2.4	PayloadTCPSocket	91
5.71.2.5	PayloadTCPSocket	91
5.72	ArcMCCTLS::PayloadTLSMCC Class Reference	92
5.72.1	Constructor & Destructor Documentation	92
5.72.1.1	PayloadTLSMCC	92
5.72.1.2	PayloadTLSMCC	92
5.72.1.3	PayloadTLSMCC	92
5.73	ArcMCCTLS::PayloadTLSSStream Class Reference	93
5.73.1	Detailed Description	93
5.73.2	Constructor & Destructor Documentation	93
5.73.2.1	PayloadTLSSStream	93
5.73.2.2	~PayloadTLSSStream	93
5.73.3	Member Function Documentation	93
5.73.3.1	GetCert	93
5.73.3.2	GetPeerCert	94
5.73.3.3	STACK_OF	94
5.73.4	Field Documentation	94
5.73.4.1	ssl_	94
5.74	ArcSec::PDPServiceInvoker Class Reference	95
5.74.1	Detailed Description	95
5.75	ArcSec::SAML2SSO_AssertionConsumerSH Class Reference	96
5.75.1	Detailed Description	96
5.76	ArcSec::SAMLTokenSH Class Reference	97
5.76.1	Detailed Description	97
5.77	ArcSec::SimpleListPDP Class Reference	98
5.77.1	Detailed Description	98
5.78	ArcSHCLegacy::SimpleMap Class Reference	99

5.79 ArcDMCSR::SRM1Client Class Reference	100
5.79.1 Member Function Documentation	100
5.79.1.1 abort	100
5.79.1.2 checkPermissions	101
5.79.1.3 copy	101
5.79.1.4 getRequestTokens	101
5.79.1.5 getSpaceTokens	101
5.79.1.6 getTURLs	102
5.79.1.7 getTURLsStatus	102
5.79.1.8 info	102
5.79.1.9 info	103
5.79.1.10 mkDir	103
5.79.1.11 ping	104
5.79.1.12 putTURLs	104
5.79.1.13 putTURLsStatus	104
5.79.1.14 release	105
5.79.1.15 releaseGet	105
5.79.1.16 releasePut	105
5.79.1.17 remove	105
5.79.1.18 rename	106
5.79.1.19 requestBringOnline	106
5.79.1.20 requestBringOnlineStatus	106
5.80 ArcDMCSR::SRM22Client Class Reference	107
5.80.1 Constructor & Destructor Documentation	107
5.80.1.1 SRM22Client	107
5.80.1.2 ~SRM22Client	107
5.80.2 Member Function Documentation	108
5.80.2.1 abort	108
5.80.2.2 checkPermissions	108
5.80.2.3 copy	108
5.80.2.4 getRequestTokens	108
5.80.2.5 getSpaceTokens	108
5.80.2.6 getTURLs	108
5.80.2.7 getTURLsStatus	108
5.80.2.8 info	109
5.80.2.9 info	109

5.80.2.10	mkDir	109
5.80.2.11	ping	109
5.80.2.12	putURLs	109
5.80.2.13	putURLsStatus	109
5.80.2.14	release	109
5.80.2.15	releaseGet	110
5.80.2.16	releasePut	110
5.80.2.17	remove	110
5.80.2.18	rename	110
5.80.2.19	requestBringOnline	110
5.80.2.20	requestBringOnlineStatus	110
5.81	ArcDMCSRMSRMClient Class Reference	111
5.81.1	Detailed Description	112
5.81.2	Constructor & Destructor Documentation	112
5.81.2.1	~SRMClient	112
5.81.3	Member Function Documentation	112
5.81.3.1	abort	112
5.81.3.2	checkPermissions	112
5.81.3.3	copy	113
5.81.3.4	getInstance	113
5.81.3.5	getRequestTokens	113
5.81.3.6	getSpaceTokens	114
5.81.3.7	getURLs	114
5.81.3.8	getURLsStatus	114
5.81.3.9	getVersion	115
5.81.3.10	info	115
5.81.3.11	info	115
5.81.3.12	mkDir	115
5.81.3.13	ping	116
5.81.3.14	putURLs	116
5.81.3.15	putURLsStatus	116
5.81.3.16	release	117
5.81.3.17	releaseGet	117
5.81.3.18	releasePut	117
5.81.3.19	remove	118
5.81.3.20	rename	118

5.81.3.21 requestBringOnline	118
5.81.3.22 requestBringOnlineStatus	118
5.81.4 Field Documentation	119
5.81.4.1 service_endpoint	119
5.82 ArcDMCSRm::SRMClientRequest Class Reference	120
5.82.1 Detailed Description	120
5.82.2 Constructor & Destructor Documentation	120
5.82.2.1 SRMClientRequest	120
5.82.2.2 SRMClientRequest	120
5.82.3 Member Function Documentation	120
5.82.3.1 waiting_time	120
5.83 ArcDMCSRm::SRMFileInfo Class Reference	121
5.83.1 Detailed Description	121
5.84 ArcDMCSRm::SRMFileMetaData Struct Reference	122
5.84.1 Detailed Description	122
5.85 ArcDMCSRm::SRMInfo Class Reference	123
5.85.1 Detailed Description	123
5.86 ArcDMCSRm::SRMInvalidRequestException Class Reference	124
5.86.1 Detailed Description	124
5.87 ArcDMCSRm::SRMURL Class Reference	125
5.87.1 Constructor & Destructor Documentation	125
5.87.1.1 SRMURL	125
5.87.2 Member Function Documentation	125
5.87.2.1 BaseURL	125
5.87.2.2 ContactURL	125
5.87.2.3 Endpoint	125
5.87.2.4 FileName	125
5.87.2.5 FullURL	125
5.87.2.6 PortDefined	125
5.87.2.7 SetSRMVersion	126
5.87.2.8 ShortURL	126
5.88 ArcDMCHTTP::StreamBuffer Class Reference	127
5.89 ArcSHCLegacy::UnixMap Class Reference	128
5.90 ArcSec::UsernameTokenSH Class Reference	129
5.90.1 Detailed Description	129
5.91 ArcSHCLegacy::voms Struct Reference	130

5.91.1 Detailed Description	130
5.91.2 Field Documentation	130
5.91.2.1 attrs	130
5.91.2.2 server	130
5.91.2.3 voname	130
5.92 ArcSHCLegacy::voms_attrs Struct Reference	131
5.92.1 Detailed Description	131
5.92.2 Field Documentation	131
5.92.2.1 cap	131
5.92.2.2 group	131
5.92.2.3 role	131
5.93 ArcSec::X509TokenSH Class Reference	132
5.93.1 Detailed Description	132
5.94 ArcSec::XACMLAlgFactory Class Reference	133
5.94.1 Detailed Description	133
5.94.2 Member Function Documentation	133
5.94.2.1 createAlg	133
5.95 ArcSec::XACMLApply Class Reference	134
5.96 ArcSec::XACMLAttributeFactory Class Reference	135
5.96.1 Detailed Description	135
5.96.2 Member Function Documentation	135
5.96.2.1 createValue	135
5.97 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference	136
5.97.1 Detailed Description	136
5.98 ArcSec::XACMLCondition Class Reference	137
5.98.1 Detailed Description	137
5.98.2 Constructor & Destructor Documentation	137
5.98.2.1 XACMLCondition	137
5.99 ArcSec::XACMLEvaluationCtx Class Reference	138
5.99.1 Detailed Description	138
5.99.2 Constructor & Destructor Documentation	138
5.99.2.1 XACMLEvaluationCtx	138
5.100 ArcSec::XACMLEvaluator Class Reference	139
5.100.1 Detailed Description	139
5.100.2 Member Function Documentation	139
5.100.2.1 evaluate	139

5.101ArcSec::XACMLFnFactory Class Reference	140
5.101.1 Detailed Description	140
5.101.2 Member Function Documentation	140
5.101.2.1 createFn	140
5.102ArcSec::XACMLPDP Class Reference	141
5.102.1 Detailed Description	141
5.103ArcSec::XACMLPolicy Class Reference	142
5.103.1 Detailed Description	142
5.103.2 Constructor & Destructor Documentation	142
5.103.2.1 XACMLPolicy	142
5.103.2.2 XACMLPolicy	142
5.103.2.3 XACMLPolicy	142
5.103.3 Member Function Documentation	142
5.103.3.1 make_policy	142
5.104ArcSec::XACMLRequest Class Reference	143
5.104.1 Member Function Documentation	143
5.104.1.1 getEvalName	143
5.104.1.2 getName	143
5.105ArcSec::XACMLRule Class Reference	144
5.105.1 Detailed Description	144
5.106ArcSec::XACMLTarget Class Reference	145
5.106.1 Detailed Description	145
5.106.2 Constructor & Destructor Documentation	145
5.106.2.1 XACMLTarget	145
5.107ArcSec::XACMLTargetMatch Class Reference	146
5.108ArcSec::XACMLTargetMatchGroup Class Reference	147
5.109ArcSec::XACMLTargetSection Class Reference	148

Chapter 1

Namespace Index

1.1 Namespace List

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

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

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

ArcSec::AllowPDP	15
ArcSec::ArcAlgFactory	16
ArcSec::ArcAttributeFactory	17
ArcSec::ArcAttributeProxy< TheAttribute >	18
ArcSec::ArcAuthZ	19
ArcSec::ArcEvaluationCtx	20
ArcSec::ArcEvaluator	21
ArcSec::ArcFnFactory	22
ArcSec::ArcPDP	23
ArcSec::ArcPolicy	24
ArcSec::ArcRequest	25
ArcSec::ArcRequestItem	26
ArcSec::ArcRequestTuple	27
ArcSec::ArcRule	28
ArcSec::AttributeDesignator	29
ArcSec::AttributeSelector	30
ArcSHCLegacy::AuthUser	31
ArcSHCLegacy::AuthVO	32
ArcSHCLegacy::ConfigParser	33
ArcMCCTLS::ConfigTlsmcc	34
ArcDMCARC::DataPointARC	35
ArcDMCFile::DataPointFile	36
ArcDMCGFAL::DataPointGFAL	37
ArcDMCGridFTP::DataPointGridFTP	38
ArcDMCHTTP::DataPointHTTP	39
ArcDMCLDAP::DataPointLDAP	40
ArcDMCLFC::DataPointLFC	41
ArcDMCMock::DataPointMock	42
ArcDMCSRm::DataPointSRM	43
ArcDMCXrootd::DataPointXrootd	44
ArcMCCTLSec::DelegationCollector	45
ArcMCCTLSec::DelegationMultiSecAttr	46
ArcSec::DelegationPDP	47

ArcMCCTLSec::DelegationSecAttr	48
ArcSec::DelegationSH	49
ArcSec::DenyPDP	50
ArcSec::GACLEvaluator	51
ArcSec::GACLPDP	52
ArcSec::GACLPolicy	53
ArcSec::GACLRequest	54
ArcDMCGFAL::GFALTransfer3rdParty	55
ArcDMCGFAL::GFALUtils	56
ArcDMCLDAP::LDAPQuery	57
ArcSHCLegacy::LegacyMap	59
ArcSHCLegacy::LegacyPDP	60
ArcSHCLegacy::LegacySecAttr	61
ArcSHCLegacy::LegacySecHandler	62
ArcDMCGridFTP::Lister	63
ArcMCCGSI::MCC_GSI_Client	64
ArcMCCGSI::MCC_GSI_Service	65
ArcMCCHTTP::MCC_HTTP	66
ArcMCCHTTP::MCC_HTTP_Client	67
ArcMCCHTTP::MCC_HTTP_Service	68
ArcMCCMsgValidator::MCC_MsgValidator	69
ArcMCCMsgValidator::MCC_MsgValidator_Service	70
ArcMCCSOAP::MCC_SOAP	71
ArcMCCSOAP::MCC_SOAP_Client	72
ArcMCCSOAP::MCC_SOAP_Service	73
ArcMCCTCP::MCC_TCP	74
ArcMCCTCP::MCC_TCP_Client	75
ArcMCCTCP::MCC_TCP_Service	76
ArcMCCTLS::MCC_TLS	77
ArcMCCTLS::MCC_TLS_Client	78
ArcMCCTLS::MCC_TLS_Service	79
ArcMCCGSI::PayloadGSISStream	80
ArcMCCHTTP::PayloadHTTP	81
ArcMCCHTTP::PayloadHTTPIn	84
ArcMCCHTTP::PayloadHTTPOut	86
ArcMCCHTTP::PayloadHTTPOutRaw	89
ArcMCCHTTP::PayloadHTTPOutStream	90
ArcMCCTCP::PayloadTCPsocket	91
ArcMCCTLS::PayloadTLSStream	93
ArcMCCTLS::PayloadTLSMCC	92
ArcSec::PDPSERVICEInvoker	95
ArcSec::SAML2SSO_AssertionConsumerSH	96
ArcSec::SAMLTokenSH	97
ArcSec::SimpleListPDP	98
ArcSHCLegacy::SimpleMap	99
ArcDMCSR::SRMClient	111
ArcDMCSR::SRM1Client	100
ArcDMCSR::SRM22Client	107
ArcDMCSR::SRMClientRequest	120
ArcDMCSR::SRMFileInfo	121
ArcDMCSR::SRMFileMetaData	122

ArcDMCSR::SRMInfo	123
ArcDMCSR::SRMInvalidRequestException	124
ArcDMCSR::SRMURL	125
ArcDMCHTTP::StreamBuffer	127
ArcSHCLegacy::UnixMap	128
ArcSec::UsernameTokenSH	129
ArcSHCLegacy::voms	130
ArcSHCLegacy::voms_attrs	131
ArcSec::X509TokenSH	132
ArcSec::XACMLAlgFactory	133
ArcSec::XACMLApply	134
ArcSec::XACMLAttributeFactory	135
ArcSec::XACMLAttributeProxy< TheAttribute >	136
ArcSec::XACMLCondition	137
ArcSec::XACMLEvaluationCtx	138
ArcSec::XACMLEvaluator	139
ArcSec::XACMLFnFactory	140
ArcSec::XACMLPDP	141
ArcSec::XACMLPolicy	142
ArcSec::XACMLRequest	143
ArcSec::XACMLRule	144
ArcSec::XACMLTarget	145
ArcSec::XACMLTargetMatch	146
ArcSec::XACMLTargetMatchGroup	147
ArcSec::XACMLTargetSection	148

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

ArcSec::AllowPDP (This PDP always return true (allow))	15
ArcSec::ArcAlgFactory (Algorithm factory class for Arc)	16
ArcSec::ArcAttributeFactory (Attribute factory class for Arc specified attributes)	17
ArcSec::ArcAttributeProxy< TheAttribute > (Arc specific AttributeProxy class)	18
ArcSec::ArcAuthZ (Tests message against list of PDPs)	19
ArcSec::ArcEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	20
ArcSec::ArcEvaluator (Execute the policy evaluation, based on the request and policy)	21
ArcSec::ArcFnFactory (Function factory class for Arc specified attributes)	22
ArcSec::ArcPDP (ArcPDP - PDP which can handle the Arc specific request and policy schema)	23
ArcSec::ArcPolicy (ArcPolicy class to parse and operate Arc specific <Policy> node)	24
ArcSec::ArcRequest	25
ArcSec::ArcRequestItem (Container, <Subjects, Actions, Objects, Contexts> tuple)	26
ArcSec::ArcRequestTuple (RequestTuple, container which includes the)	27
ArcSec::ArcRule (ArcRule class to parse Arc specific <Rule> node)	28
ArcSec::AttributeDesignator	29
ArcSec::AttributeSelector	30
ArcSHCLegacy::AuthUser	31
ArcSHCLegacy::AuthVO	32
ArcSHCLegacy::ConfigParser	33
ArcMCCTL::ConfigTSMCC	34
ArcDMCARC::DataPointARC	35
ArcDMCFile::DataPointFile	36
ArcDMCGFAL::DataPointGFAL	37
ArcDMCGridFTP::DataPointGridFTP	38
ArcDMCHTTP::DataPointHTTTP	39
ArcDMCLDAP::DataPointLDAP	40
ArcDMCLFC::DataPointLFC	41
ArcDMCMock::DataPointMock (Mock data point which does not do anything but sleep for each operation)	42
ArcDMCSR::DataPointSRM	43
ArcDMCXrootd::DataPointXrootd	44
ArcMCCTLSSec::DelegationCollector	45

ArcMCCTLSec::DelegationMultiSecAttr	46
ArcSec::DelegationPDP	47
ArcMCCTLSec::DelegationSecAttr	48
ArcSec::DelegationSH	49
ArcSec::DenyPDP (This PDP always returns false (deny))	50
ArcSec::GACLEvaluator	51
ArcSec::GACLPDP	52
ArcSec::GACLPolicy	53
ArcSec::GACLRequest	54
ArcDMCGFAL::GFALTransfer3rdParty (Class to interact with GFAL2 to do third-party transfer)	55
ArcDMCGFAL::GFALUtils (Utility functions for GFAL2)	56
ArcDMCLDAP::LDAPQuery	57
ArcSHCLegacy::LegacyMap	59
ArcSHCLegacy::LegacyPDP	60
ArcSHCLegacy::LegacySecAttr	61
ArcSHCLegacy::LegacySecHandler	62
ArcDMCGridFTP::Lister	63
ArcMCCGSI::MCC_GSI_Client	64
ArcMCCGSI::MCC_GSI_Service	65
ArcMCCHTTP::MCC_HTTP (A base class for HTTP client and service MCCs)	66
ArcMCCHTTP::MCC_HTTP_Client	67
ArcMCCHTTP::MCC_HTTP_Service	68
ArcMCCMsgValidator::MCC_MsgValidator	69
ArcMCCMsgValidator::MCC_MsgValidator_Service	70
ArcMCCSOAP::MCC_SOAP (A base class for SOAP client and service MCCs)	71
ArcMCCSOAP::MCC_SOAP_Client	72
ArcMCCSOAP::MCC_SOAP_Service	73
ArcMCCTCP::MCC_TCP (A base class for TCP client and service MCCs)	74
ArcMCCTCP::MCC_TCP_Client	75
ArcMCCTCP::MCC_TCP_Service	76
ArcMCCTLS::MCC_TLS (A base class for TLS client and service MCCs)	77
ArcMCCTLS::MCC_TLS_Client	78
ArcMCCTLS::MCC_TLS_Service	79
ArcMCCGSI::PayloadGSISStream	80
ArcMCCHTTP::PayloadHTTP	81
ArcMCCHTTP::PayloadHTTPIn	84
ArcMCCHTTP::PayloadHTTPOut	86
ArcMCCHTTP::PayloadHTTPOutRaw	89
ArcMCCHTTP::PayloadHTTPOutStream	90
ArcMCCTCP::PayloadTCPsocket	91
ArcMCCTLS::PayloadTLMCC	92
ArcMCCTLS::PayloadTLSSStream	93
ArcSec::PDPSERVICEInvoker (PDPSERVICEInvoker - client which will invoke pdpservice)	95
ArcSec::SAML2SSO_AssertionConsumerSH (Implement the functionality of the Service Provider in SAML2 SSO profile)	96
ArcSec::SAMLTokenSH (Adds WS-Security SAML Token into SOAP Header)	97
ArcSec::SimpleListPDP (Tests X509 subject against list of subjects in file)	98
ArcSHCLegacy::SimpleMap	99
ArcDMCSRm::SRM1Client	100
ArcDMCSRm::SRM22Client	107
ArcDMCSRm::SRMClient	111
ArcDMCSRm::SRMClientRequest (Class to represent a SRM request)	120
ArcDMCSRm::SRMFileInfo	121
ArcDMCSRm::SRMFileMetaData (SRM-related file metadata)	122

ArcDMCSRMSRMInfo	123
ArcDMCSRMSRMInvalidRequestException (General exception to represent a bad SRM request)	124
ArcDMCSRMSRMURL	125
ArcDMCHTTPStreamBuffer	127
ArcSHCLegacyUnixMap	128
ArcSecUsernameTokenSH (Adds WS-Security Username Token into SOAP Header)	129
ArcSHCLegacyvoms	130
ArcSHCLegacyvoms_attrs	131
ArcSecX509TokenSH (Adds WS-Security X509 Token into SOAP Header)	132
ArcSecXACMLAlgFactory (Algorithm factory class for XACML)	133
ArcSecXACMLApply	134
ArcSecXACMLAttributeFactory (Attribute factory class for XACML specified attributes)	135
ArcSecXACMLAttributeProxy TheAttribute (XACML specific AttributeProxy class)	136
ArcSecXACMLCondition (XACMLCondition class to parse and operate XACML specific <Condition> node)	137
ArcSecXACMLEvaluationCtx (EvaluationCtx, in charge of storing some context information for evaluation, including Request, current time, etc)	138
ArcSecXACMLEvaluator (Execute the policy evaluation, based on the request and policy)	139
ArcSecXACMLFnFactory (Function factory class for XACML specified attributes)	140
ArcSecXACMLPDP (XACMLPDP - PDP which can handle the XACML specific request and policy schema)	141
ArcSecXACMLPolicy (XACMLPolicy class to parse and operate XACML specific <Policy> node)	142
ArcSecXACMLRequest	143
ArcSecXACMLRule (XACMLRule class to parse XACML specific <Rule> node)	144
ArcSecXACMLTarget (XACMLTarget class to parse and operate XACML specific <Target> node)	145
ArcSecXACMLTargetMatch	146
ArcSecXACMLTargetMatchGroup	147
ArcSecXACMLTargetSection	148

Chapter 4

Namespace Documentation

4.1 ArcSec Namespace Reference

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

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 meant 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>                            <SubFraction
type="X500DN">/O=Grid/OU=KnowARC/CN=XYZ</SubFraction>                            <SubFraction
type="ShibName">urn:mace:shibboleth:examples</SubFraction>                        </Subject>      <GroupIdRef
location="/subjectgroup.xml">subgrpexample1</GroupIdRef> </Subjects>

```

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

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

Chapter 5

Data Structure Documentation

5.1 ArcSec::AllowPDP Class Reference

This PDP always return true (allow).

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

5.1.1 Detailed Description

This PDP always return true (allow).

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

- AllowPDP.h

5.2 ArcSec::ArcAlgFactory Class Reference

Algorithm factory class for Arc.

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

Public Member Functions

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

5.2.1 Detailed Description

Algorithm factory class for Arc.

5.2.2 Member Function Documentation

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

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

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

- ArcAlgFactory.h

5.3 ArcSec::ArcAttributeFactory Class Reference

Attribute factory class for Arc specified attributes.

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

Public Member Functions

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

5.3.1 Detailed Description

Attribute factory class for Arc specified attributes.

5.3.2 Member Function Documentation

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

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

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

- ArcAttributeFactory.h

5.4 ArcSec::ArcAttributeProxy< TheAttribute > Class Template Reference

Arc specific AttributeProxy class.

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

Public Member Functions

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

5.4.1 Detailed Description

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

Arc specific AttributeProxy class.

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

- ArcAttributeProxy.h

5.5 ArcSec::ArcAuthZ Class Reference

Tests message against list of PDPs.

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

Data Structures

- class **PDPDesc**

Public Member Functions

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

Protected Member Functions

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

5.5.1 Detailed Description

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

5.5.2 Member Function Documentation

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

Get authorization decision

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

Create PDP according to conf info

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

- ArcAuthZ.h

5.6 ArcSec::ArcEvaluationCtx Class Reference

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

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

Public Member Functions

- [ArcEvaluationCtx](#) (Request *request)
- virtual void [split](#) ()

5.6.1 Detailed Description

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

5.6.2 Constructor & Destructor Documentation

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

Construct a new EvaluationCtx based on the given request

5.6.3 Member Function Documentation

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

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

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

- ArcEvaluationCtx.h

5.7 ArcSec::ArcEvaluator Class Reference

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

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

Public Member Functions

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

5.7.1 Detailed Description

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

5.7.2 Member Function Documentation

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

Evaluate the request based on the policy information inside PolicyStore

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

- ArcEvaluator.h

5.8 ArcSec::ArcFnFactory Class Reference

Function factory class for Arc specified attributes.

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

Public Member Functions

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

5.8.1 Detailed Description

Function factory class for Arc specified attributes.

5.8.2 Member Function Documentation

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

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

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

- ArcFnFactory.h

5.9 ArcSec::ArcPDP Class Reference

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

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

5.9.1 Detailed Description

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

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

- ArcPDP.h

5.10 ArcSec::ArcPolicy Class Reference

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

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

Public Member Functions

- [ArcPolicy](#) (Arc::PluginArgument *parg)
- [ArcPolicy](#) (const Arc::XMLNode node, Arc::PluginArgument *parg)
- [ArcPolicy](#) (const Arc::XMLNode node, EvaluatorContext *ctx, Arc::PluginArgument *parg)
- 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 (Arc::PluginArgument * *parg*)

Constructor

5.10.2.2 ArcSec::ArcPolicy::ArcPolicy (const Arc::XMLNode *node*, Arc::PluginArgument * *parg*)

Constructor

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

Constructor

5.10.3 Member Function Documentation

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

Parse XMLNode, and construct the low-level Rule object

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

- ArcPolicy.h

5.11 ArcSec::ArcRequest Class Reference

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

- ArcRequest.h

5.12 ArcSec::ArcRequestItem Class Reference

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

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

5.12.1 Detailed Description

Container, <Subjects, Actions, Objects, Contexts> tuple. Specified [ArcRequestItem](#) which can parse Arc request formate

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

- ArcRequestItem.h

5.13 ArcSec::ArcRequestTuple Class Reference

RequestTuple, container which includes the.

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

5.13.1 Detailed Description

RequestTuple, container which includes the.

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

- ArcEvaluationCtx.h

5.14 ArcSec::ArcRule Class Reference

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

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

5.14.1 Detailed Description

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

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

- ArcRule.h

5.15 ArcSec::AttributeDesignator Class Reference

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

- AttributeDesignator.h

5.16 ArcSec::AttributeSelector Class Reference

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

- AttributeSelector.h

5.17 ArcSHCLegacy::AuthUser Class Reference

Data Structures

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

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

- `auth.h`

5.18 ArcSHCLegacy::AuthVO Class Reference

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

- auth.h

5.19 ArcSHCLegacy::ConfigParser Class Reference

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

- ConfigParser.h

5.20 ArcMCCTLS::ConfigTLMCC Class Reference

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

- ConfigTLMCC.h

5.21 ArcDMCARC::DataPointARC Class Reference

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

5.21.1 Detailed Description

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

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

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

- DataPointARC.h

5.22 ArcDMCFile::DataPointFile Class Reference

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

5.22.1 Detailed Description

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

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

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

- DataPointFile.h

5.23 ArcDMCGFAL::DataPointGFAL Class Reference

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

5.23.1 Detailed Description

Provides access to the gLite Grid File Access Library through ARC's API. The following protocols are supported: lfc, srm, root, gsiftp, rfio, dcap and gsidcap.

Notes on env variables:

- If SRM is used LCG_GFAL_INFOSYS must be set to BDII host:port unless the full URL with port and web service path is given.

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

5.24 ArcDMCGridFTP::DataPointGridFTP Class Reference

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

Data Structures

- class **CBArg**

5.24.1 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

5.25 ArcDMCHTTP::DataPointHTTP Class Reference

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

5.25.1 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

5.26 ArcDMCLDAP::DataPointLDAP Class Reference

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

5.26.1 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

5.27 ArcDMCLFC::DataPointLFC Class Reference

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

5.27.1 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

5.28 ArcDMCMock::DataPointMock Class Reference

Mock data point which does not do anything but sleep for each operation.

```
#include <DataPointMock.h>
```

5.28.1 Detailed Description

Mock data point which does not do anything but sleep for each operation. If the URL protocol is mock:// then each method returns `DataStatus::Success`. If it is fail:// then each method returns an error. This plugin is not built by default - to build it the option `--enable-mock-dmc` must be passed to configure.

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

- `DataPointMock.h`

5.29 ArcDMCSRMR::DataPointSRM Class Reference

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

5.29.1 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

5.30 ArcDMCXrootd::DataPointXrootd Class Reference

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

5.30.1 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

5.31 ArcMCCTLSec::DelegationCollector Class Reference

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

- DelegationCollector.h

5.32 ArcMCCTLSec::DelegationMultiSecAttr Class Reference

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

- DelegationSecAttr.h

5.33 ArcSec::DelegationPDP Class Reference

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

5.33.1 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

5.34 ArcMCCTLSec::DelegationSecAttr Class Reference

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

- DelegationSecAttr.h

5.35 ArcSec::DelegationSH Class Reference

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

- DelegationSH.h

5.36 ArcSec::DenyPDP Class Reference

This PDP always returns false (deny).

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

5.36.1 Detailed Description

This PDP always returns false (deny).

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

- DenyPDP.h

5.37 ArcSec::GACLEvaluator Class Reference

Public Member Functions

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

5.37.1 Member Function Documentation

5.37.1.1 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

5.38 ArcSec::GACLPDP Class Reference

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

- GACLPDP.h

5.39 ArcSec::GACLPolicy Class Reference

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

- GACLPolicy.h

5.40 ArcSec::GACLRequest Class Reference

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

- GACLRequest.h

5.41 ArcDMCGFAL::GFALTransfer3rdParty Class Reference

Class to interact with GFAL2 to do third-party transfer.

```
#include <GFALTransfer3rdParty.h>
```

Public Member Functions

- [GFALTransfer3rdParty](#) (const URL &source, const URL &dest, const Arc::UserConfig &cfg, DataPoint::Callback3rdParty cb)
- DataStatus [Transfer](#) ()

5.41.1 Detailed Description

Class to interact with GFAL2 to do third-party transfer.

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

- GFALTransfer3rdParty.h

5.42 ArcDMCGFAL::GFALUtils Class Reference

Utility functions for GFAL2.

```
#include <GFALUtils.h>
```

Static Public Member Functions

- static std::string [GFALURL](#) (const URL &u)
- static int [HandleGFALError](#) (Logger &logger)

5.42.1 Detailed Description

Utility functions for GFAL2.

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

- GFALUtils.h

5.43 ArcDMCLDAP::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](#) ()
- int [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)
- int [Result](#) (ldap_callback callback, void *ref)

5.43.1 Detailed Description

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

5.43.2 Constructor & Destructor Documentation

5.43.2.1 ArcDMCLDAP::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](#).

5.43.2.2 ArcDMCLDAP::LDAPQuery::~~LDAPQuery ()

Destructor. Will disconnect from the ldapserver if still connected.

5.43.3 Member Function Documentation

5.43.3.1 int ArcDMCLDAP::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.

Returns:

0: success, 1: timeout, -1: other error

5.43.3.2 int ArcDMCLDAP::LDAPQuery::Result (ldap_callback callback, void *ref)

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

Returns:

0: success, 1: timeout, -1: other error

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

- LDAPQuery.h

5.44 ArcSHCLegacy::LegacyMap Class Reference

Data Structures

- class `cfgfile`

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

- `LegacyMap.h`

5.45 ArcSHCLegacy::LegacyPDP Class Reference

Data Structures

- class **cfgfile**

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

- LegacyPDP.h

5.46 ArcSHCLegacy::LegacySecAttr Class Reference

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

- LegacySecAttr.h

5.47 ArcSHCLegacy::LegacySecHandler Class Reference

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

- LegacySecHandler.h

5.48 ArcDMCGridFTP::Lister Class Reference

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

- Lister.h

5.49 ArcMCCGSI::MCC_GSI_Client Class Reference

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

- MCCGSI.h

5.50 ArcMCCGSI::MCC_GSI_Service Class Reference

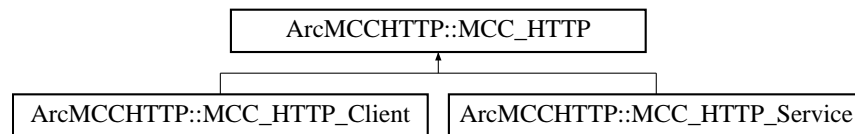
The documentation for this class was generated from the following file:

- MCCGSI.h

5.51 ArcMCCHTTP::MCC_HTTP Class Reference

A base class for HTTP client and service MCCs.

`#include <MCCHTTP.h>` Inheritance diagram for ArcMCCHTTP::MCC_HTTP::



5.51.1 Detailed Description

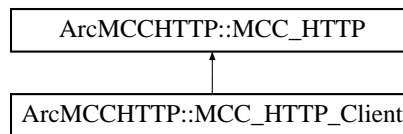
A base class for HTTP client and service MCCs. This is a base class for HTTP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

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

- MCCHTTP.h

5.52 ArcMCCHTTP::MCC_HTTP_Client Class Reference

#include <MCCHTTP.h> Inheritance diagram for ArcMCCHTTP::MCC_HTTP_Client::



5.52.1 Detailed Description

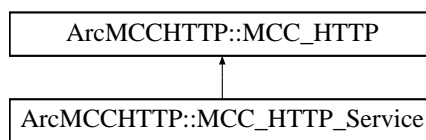
This class is a client part of HTTP MCC. It accepts PayloadRawInterface payload and uses it as body to generate HTTP request. Request is passed to next MCC as PayloadRawInterface type of payload. Returned PayloadStreamInterface payload is parsed into HTTP response and it's body is passed back to calling 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

5.53 ArcMCCHTTP::MCC_HTTP_Service Class Reference

#include <MCCHTTP.h> Inheritance diagram for ArcMCCHTTP::MCC_HTTP_Service::



5.53.1 Detailed Description

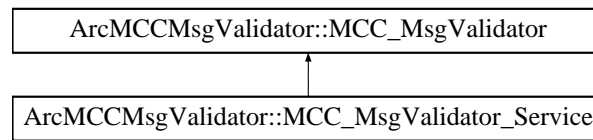
This class implements MCC to processes HTTP request. On input payload with PayloadStreamInterface is expected. HTTP message is read from stream and its body is converted into PayloadRaw and passed to next MCC. Returned payload of PayloadRawInterface type is treated as body part of returning [PayloadHTTP](#).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

5.54 ArcMCCMsgValidator::MCC_MsgValidator Class Reference

Inheritance diagram for ArcMCCMsgValidator::MCC_MsgValidator::

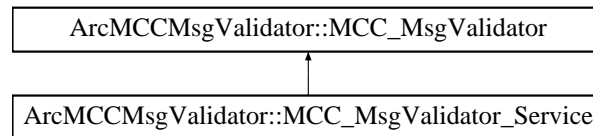


The documentation for this class was generated from the following file:

- MCCMsgValidator.h

5.55 ArcMCCMsgValidator::MCC_MsgValidator_Service Class Reference

Inheritance diagram for ArcMCCMsgValidator::MCC_MsgValidator_Service::



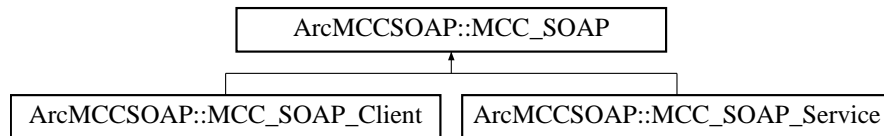
The documentation for this class was generated from the following file:

- MCCMsgValidator.h

5.56 ArcMCCSOAP::MCC_SOAP Class Reference

A base class for SOAP client and service MCCs.

#include <MCCSOAP.h> Inheritance diagram for ArcMCCSOAP::MCC_SOAP::



5.56.1 Detailed Description

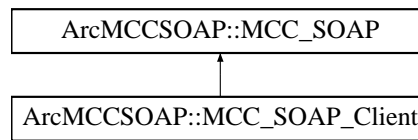
A base class for SOAP client and service MCCs. This is a base class for SOAP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

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

- MCCSOAP.h

5.57 ArcMCCSOAP::MCC_SOAP_Client Class Reference

Inheritance diagram for ArcMCCSOAP::MCC_SOAP_Client::

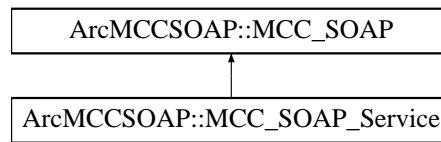


The documentation for this class was generated from the following file:

- MCCSOAP.h

5.58 ArcMCCSOAP::MCC_SOAP_Service Class Reference

#include <MCCSOAP.h> Inheritance diagram for ArcMCCSOAP::MCC_SOAP_Service::



5.58.1 Detailed Description

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

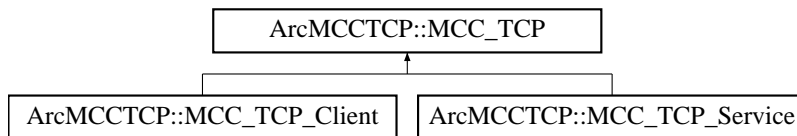
The documentation for this class was generated from the following file:

- MCCSOAP.h

5.59 ArcMCCTCP::MCC_TCP Class Reference

A base class for TCP client and service MCCs.

`#include <MCCTCP.h>` Inheritance diagram for ArcMCCTCP::MCC_TCP::



5.59.1 Detailed Description

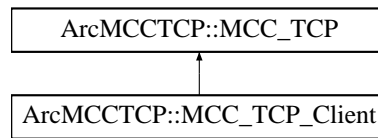
A base class for TCP client and service MCCs. This is a base class for TCP client and service MCCs. It provides some common functionality for them, i.e. so far only a logger.

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

- MCCTCP.h

5.60 ArcMCCTCP::MCC_TCP_Client Class Reference

`#include <MCCTCP.h>`Inheritance diagram for ArcMCCTCP::MCC_TCP_Client::



5.60.1 Detailed Description

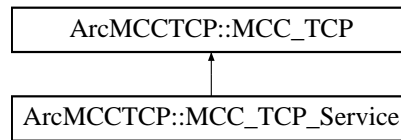
This class is MCC implementing TCP client. Upon creation it connects to specified TCP post at specified host. `process()` method accepts `PayloadRawInterface` type of payload. Content of payload is sent over TCP socket. It returns `PayloadStreamInterface` payload for previous MCC to read response.

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

- MCCTCP.h

5.61 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, PluginArgument *parg)

5.61.1 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

5.61.2 Constructor & Destructor Documentation

5.61.2.1 ArcMCCTCP::MCC_TCP_Service::MCC_TCP_Service (Config *cfg, PluginArgument *parg)

executing function for connection thread

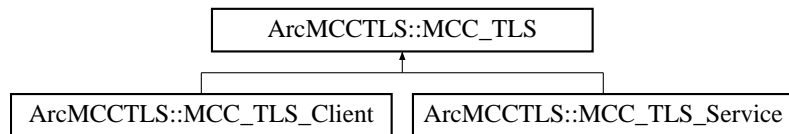
The documentation for this class was generated from the following file:

- MCCTCP.h

5.62 ArcMCCTLS::MCC_TLS Class Reference

A base class for TLS client and service MCCs.

`#include <MCCTLS.h>`Inheritance diagram for ArcMCCTLS::MCC_TLS::



5.62.1 Detailed Description

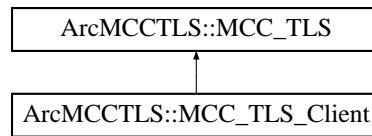
A base class for TLS client and service MCCs. This is a base class for TLS client and service MCCs. It provides some common functionality for them.

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

- MCCTLS.h

5.63 ArcMCCTLS::MCC_TLS_Client Class Reference

`#include <MCCTLS.h>`Inheritance diagram for ArcMCCTLS::MCC_TLS_Client::



5.63.1 Detailed Description

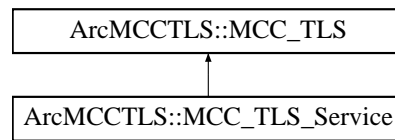
This class is MCC implementing TLS client.

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

- MCCTLS.h

5.64 ArcMCCTLS::MCC_TLS_Service Class Reference

#include <MCCTLS.h> Inheritance diagram for ArcMCCTLS::MCC_TLS_Service::



5.64.1 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 creat socket in the constructor method by using information in configuration file), we can only creat "ssl" object which is binded to specified "socket", when MCC_HTTP_Client 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

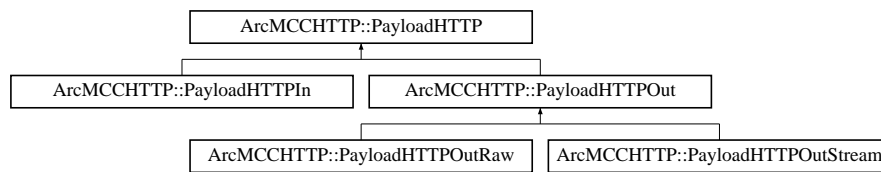
5.65 ArcMCCGSI::PayloadGSISStream Class Reference

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

- PayloadGSISStream.h

5.66 ArcMCCHTTP::PayloadHTTP Class Reference

#include <PayloadHTTP.h> Inheritance diagram for ArcMCCHTTP::PayloadHTTP::



Public Member Functions

- [PayloadHTTP](#) (void)
- [PayloadHTTP](#) (const std::string &method, const std::string &url)
- [PayloadHTTP](#) (int code, const std::string &reason)
- virtual const std::string & [Attribute](#) (const std::string &name)
- virtual const std::multimap< std::string, std::string > & [Attributes](#) (void)
- std::string [GetError](#) ()

Protected Attributes

- std::string [uri_](#)
- int [version_major_](#)
- int [version_minor_](#)
- std::string [method_](#)
- int [code_](#)
- std::string [reason_](#)
- int64_t [length_](#)
- int64_t [offset_](#)
- int64_t [size_](#)
- int64_t [end_](#)
- bool [keep_alive_](#)
- std::multimap< std::string, std::string > [attributes_](#)

5.66.1 Detailed Description

These classes implement parsing and generation of HTTP messages. They implement only subset of HTTP/1.1 and also provides PayloadRawInterface and PayloadStreamInterface for including as payload into Message passed through MCC chains.

5.66.2 Constructor & Destructor Documentation

5.66.2.1 ArcMCCHTTP::PayloadHTTP::PayloadHTTP (void)

Textual description of error which happened Constructor - creates empty object

5.66.2.2 ArcMCCHTTP::PayloadHTTTP::PayloadHTTTP (const std::string & *method*, const std::string & *url*)

Constructor - creates HTTP request.

5.66.2.3 ArcMCCHTTP::PayloadHTTTP::PayloadHTTTP (int *code*, const std::string & *reason*)

Constructor - creates HTTP response.

5.66.3 Member Function Documentation

5.66.3.1 virtual const std::string& ArcMCCHTTP::PayloadHTTTP::Attribute (const std::string & *name*) [virtual]

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

5.66.3.2 virtual const std::multimap<std::string, std::string> & ArcMCCHTTP::PayloadHTTTP::Attributes (void) [virtual]

Returns all HTTP header attributes.

5.66.3.3 std::string ArcMCCHTTP::PayloadHTTTP::GetError () [inline]

Returns textual description of last error

5.66.4 Field Documentation

5.66.4.1 std::multimap<std::string, std::string> ArcMCCHTTP::PayloadHTTTP::attributes_ [protected]

true if connection should not be closed after response

5.66.4.2 int ArcMCCHTTP::PayloadHTTTP::code_ [protected]

HTTP method being used or requested (if request)

5.66.4.3 int64_t ArcMCCHTTP::PayloadHTTTP::end_ [protected]

Logical size of content obtained from Content-Range

5.66.4.4 bool ArcMCCHTTP::PayloadHTTTP::keep_alive_ [protected]

Logical end of content computed from Content-Range

5.66.4.5 int64_t ArcMCCHTTP::PayloadHTTTP::length_ [protected]

HTTP reason being sent or supplied (if response)

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

minor number of HTTP version - must be 0 or 1

5.66.4.7 `int64_t ArcMCCHTTP::PayloadHTTP::offset_` `[protected]`

Content-length of HTTP message

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

HTTP code being sent or supplied (if response)

5.66.4.9 `int64_t ArcMCCHTTP::PayloadHTTP::size_` `[protected]`

Logical beginning of content computed from Content-Range

5.66.4.10 `std::string ArcMCCHTTP::PayloadHTTP::uri_` `[protected]`

set to true if object is valid

5.66.4.11 `int ArcMCCHTTP::PayloadHTTP::version_major_` `[protected]`

URI being contacted

5.66.4.12 `int ArcMCCHTTP::PayloadHTTP::version_minor_` `[protected]`

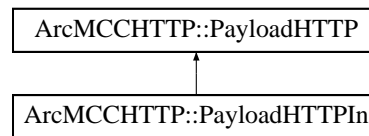
major number of HTTP version - must be 1

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

- PayloadHTTP.h

5.67 ArcMCCHTTP::PayloadHTTPIn Class Reference

Inheritance diagram for ArcMCCHTTP::PayloadHTTPIn::



Public Member Functions

- [PayloadHTTPIn](#) (PayloadStreamInterface &stream, bool own=false)

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

- int64_t [chunk_size_](#)
- uint64_t [stream_offset_](#)
- bool [stream_own_](#)
- bool [fetched_](#)
- char [tbuf_](#) [1024]
- int [tbuflen_](#)
- char * [body_](#)

5.67.1 Constructor & Destructor Documentation

5.67.1.1 ArcMCCHTTP::PayloadHTTPIn::PayloadHTTPIn (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.

5.67.2 Member Function Documentation

5.67.2.1 bool ArcMCCHTTP::PayloadHTTPIn::get_body (void) [protected]

Read Body of HTTP message and store it into internal buffer. Avoid using it unless really needed because it can read a lot of data. TODO: maybe using on disk buffer can help with GB sized bodies.

5.67.2.2 bool ArcMCCHTTP::PayloadHTTPIn::read (char * *buf*, int64_t & *size*) [protected]

Read up to 'size' bytes from stream_

5.67.2.3 bool ArcMCCHTTP::PayloadHTTPIn::read_header (void) [protected]

Read HTTP header and fill internal variables

5.67.2.4 bool ArcMCCHTTP::PayloadHTTPIn::readline (std::string & *line*) [protected]

Read from stream_ till

5.67.3 Field Documentation**5.67.3.1 char* ArcMCCHTTP::PayloadHTTPIn::body_ [protected]**

amount of data stored in tbuf

5.67.3.2 int64_t ArcMCCHTTP::PayloadHTTPIn::chunk_size_ [protected]

chunked encoding parsing state

5.67.3.3 bool ArcMCCHTTP::PayloadHTTPIn::fetched_ [protected]

if true stream_ is owned by this

5.67.3.4 uint64_t ArcMCCHTTP::PayloadHTTPIn::stream_offset_ [protected]

stream used to pull HTTP data from

5.67.3.5 bool ArcMCCHTTP::PayloadHTTPIn::stream_own_ [protected]

amount of data read from stream_

5.67.3.6 char ArcMCCHTTP::PayloadHTTPIn::tbuf_[1024] [protected]

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

5.67.3.7 int ArcMCCHTTP::PayloadHTTPIn::tbuflen_ [protected]

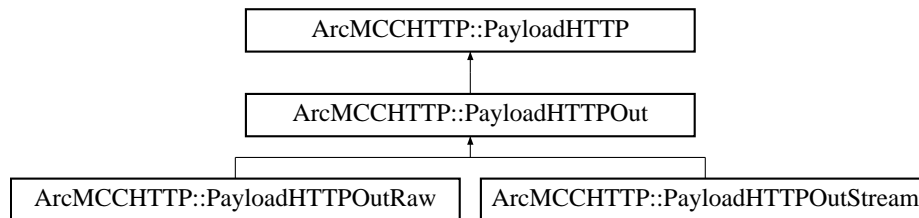
intermediate buffer for reading header lines

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

- PayloadHTTP.h

5.68 ArcMCCHTTP::PayloadHTTPOut Class Reference

Inheritance diagram for ArcMCCHTTP::PayloadHTTPOut::



Public Member Functions

- [PayloadHTTPOut](#) (const std::string &method, const std::string &url)
- [PayloadHTTPOut](#) (int code, const std::string &reason, bool head_response=false)
- virtual void [Attribute](#) (const std::string &name, const std::string &value)
- virtual bool [Flush](#) (PayloadStreamInterface &stream)

Protected Member Functions

- bool [make_header](#) (bool to_stream)

Protected Attributes

- PayloadRawInterface * [rbody_](#)
- PayloadStreamInterface * [sbody_](#)
- PayloadStreamInterface::Size_t [sbody_size_](#)
- std::string [header_](#)
- bool [to_stream_](#)
- bool [use_chunked_transfer_](#)
- uint64_t [stream_offset_](#)

5.68.1 Constructor & Destructor Documentation

5.68.1.1 ArcMCCHTTP::PayloadHTTPOut::PayloadHTTPOut (const std::string & *method*, const std::string & *url*)

Constructor - creates HTTP request to be sent.

5.68.1.2 ArcMCCHTTP::PayloadHTTPOut::PayloadHTTPOut (int *code*, const std::string & *reason*, bool *head_response* = **false**)

Constructor - creates HTTP response to be sent. If 'head_response' is set to true then response is generated as if it is result of HEAD request.

5.68.2 Member Function Documentation

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

Adds HTTP header attribute 'name' = 'value'

5.68.2.2 `virtual bool ArcMCCHTTP::PayloadHTTPOut::Flush (PayloadStreamInterface & stream) [virtual]`

Send created object through provided stream.

5.68.2.3 `bool ArcMCCHTTP::PayloadHTTPOut::make_header (bool to_stream) [protected]`

Amount of data read read through Stream interface

5.68.3 Field Documentation

5.68.3.1 `std::string ArcMCCHTTP::PayloadHTTPOut::header_ [protected]`

if true rbody_ and sbody_ is owned by this

5.68.3.2 `PayloadRawInterface* ArcMCCHTTP::PayloadHTTPOut::rbody_ [protected]`

true if HTTP response for HEAD request to be generated

5.68.3.3 `PayloadStreamInterface* ArcMCCHTTP::PayloadHTTPOut::sbody_ [protected]`

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

5.68.3.4 `PayloadStreamInterface::Size_t ArcMCCHTTP::PayloadHTTPOut::sbody_size_ [protected]`

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

5.68.3.5 `uint64_t ArcMCCHTTP::PayloadHTTPOut::stream_offset_ [protected]`

Chunked transfer to be used

5.68.3.6 `bool ArcMCCHTTP::PayloadHTTPOut::to_stream_ [protected]`

Header to be prepended to body

5.68.3.7 `bool ArcMCCHTTP::PayloadHTTPOut::use_chunked_transfer_ [protected]`

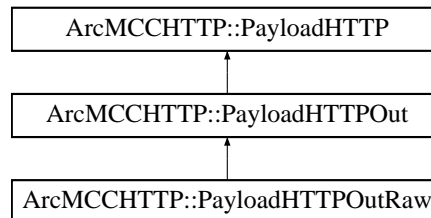
Header was generated for streaming data

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

- PayloadHTTP.h

5.69 ArcMCCHTTP::PayloadHTTPOutRaw Class Reference

Inheritance diagram for ArcMCCHTTP::PayloadHTTPOutRaw::



Public Member Functions

- virtual void [Body](#) (PayloadRawInterface &body, bool ownership=true)

5.69.1 Member Function Documentation

5.69.1.1 virtual void ArcMCCHTTP::PayloadHTTPOutRaw::Body (PayloadRawInterface &body, bool ownership = true) [virtual]

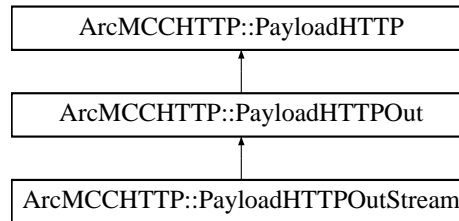
Assign HTTP body. Assigned object is not copied. Instead it is remembered and made available through Raw and Stream interfaces. Previously attached body is discarded. If 'ownership' is true then passed object is treated as being owned by this instance and destroyed in destructor or when discarded.

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

- PayloadHTTP.h

5.70 ArcMCCHTTP::PayloadHTTPOutStream Class Reference

Inheritance diagram for ArcMCCHTTP::PayloadHTTPOutStream::



Public Member Functions

- virtual void [Body](#) (PayloadStreamInterface &body, bool ownership=true)

5.70.1 Member Function Documentation

5.70.1.1 virtual void ArcMCCHTTP::PayloadHTTPOutStream::Body (PayloadStreamInterface & *body*, bool *ownership* = true) [virtual]

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

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]

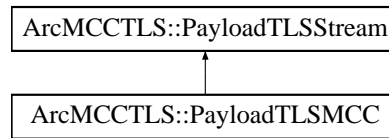
Copy constructor - inherits handle of copied object. Handle is NOT closed in destructor.

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*)

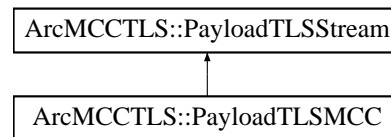
Copy constructor with new logger. Created object shares same SSL objects but does not destroy them in destructor. Main instance must be destroyed after all copied ones.

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

Implementation 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::PDPServiceInvoker Class Reference

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

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

5.74.1 Detailed Description

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

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

- PDPServiceInvoker.h

5.75 ArcSec::SAML2SSO_AssertionConsumerSH Class Reference

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

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

5.75.1 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

5.76 ArcSec::SAMLTokenSH Class Reference

Adds WS-Security SAML Token into SOAP Header.

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

5.76.1 Detailed Description

Adds WS-Security SAML Token into SOAP Header.

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

- SAMLTokenSH.h

5.77 ArcSec::SimpleListPDP Class Reference

Tests X509 subject against list of subjects in file.

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

5.77.1 Detailed Description

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

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

- SimpleListPDP.h

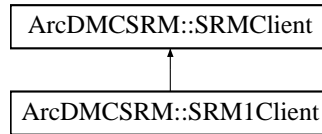
5.78 ArcSHCLegacy::SimpleMap Class Reference

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

- simplemap.h

5.79 ArcDMCSRMSRMClient Class Reference

Inheritance diagram for ArcDMCSRMSRMClient:



Public Member Functions

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

5.79.1 Member Function Documentation

5.79.1.1 DataStatus ArcDMCSRMSRMClient::abort (SRMClientRequest & req, bool source) [virtual]

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

Parameters:

- req* The request object
source Whether source or destination is being aborted

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

description The space token description

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRm::SRMClient](#).

5.79.1.6 DataStatus ArcDMCSRm::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:

req The request object

urls A list of TURLs filled by the method

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRm::SRMClient](#).

5.79.1.7 DataStatus ArcDMCSRm::SRM1Client::getTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [inline, virtual]

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

Parameters:

req The request object. Status must be ongoing.

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

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRm::SRMClient](#).

5.79.1.8 DataStatus ArcDMCSRm::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 ≥ 1 then the list in metadata will contain the content of the directory.

Parameters:

req The request object
metadata A list of structs filled with file information.

Returns:

DataStatus specifying outcome of operation

See also:

[SRMFileMetaData](#)

Implements [ArcDMCSRM::SRMClient](#).

5.79.1.9 DataStatus ArcDMCSRM::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:

req The request object
metadata 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:

DataStatus specifying outcome of operation

See also:

[SRMFileMetaData](#)

Implements [ArcDMCSRM::SRMClient](#).

5.79.1.10 DataStatus ArcDMCSRM::SRM1Client::mkDir (SRMClientRequest & *req*) [inline, virtual]

Make required directories for the SURL in the request

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRM::SRMClient](#).

5.79.1.11 `DataStatus ArcDMCSRm::SRM1Client::ping (std::string & version) [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:

version The version returned by the server

Returns:

`DataStatus` specifying outcome of operation

Implements [ArcDMCSRm::SRMClient](#).

5.79.1.12 `DataStatus ArcDMCSRm::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 URLs are ready, in the latter case it will return after making the request and `putURLsStatus()` must be used to poll the request status if it was not completed.

Parameters:

req The request object

urls A list of URLs filled by the method

Returns:

`DataStatus` specifying outcome of operation

Implements [ArcDMCSRm::SRMClient](#).

5.79.1.13 `DataStatus ArcDMCSRm::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:

req The request object. Status must be ongoing.

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

Returns:

`DataStatus` specifying outcome of operation

Implements [ArcDMCSRm::SRMClient](#).

5.79.1.14 DataStatus ArcDMCSRMSRMClient::release (SRMClientRequest & *req*, bool *source*) [virtual**]**

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

Parameters:

req The request object
source Whether source or destination is being released

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

5.79.1.15 DataStatus ArcDMCSRMSRMClient::releaseGet (SRMClientRequest & *req*) [virtual**]**

Should be called after a successful copy from SRM storage.

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

5.79.1.16 DataStatus ArcDMCSRMSRMClient::releasePut (SRMClientRequest & *req*) [virtual**]**

Should be called after a successful copy to SRM storage.

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

5.79.1.17 DataStatus ArcDMCSRMSRMClient::remove (SRMClientRequest & *req*) [virtual**]**

Delete a file physically from storage and the SRM namespace.

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

5.79.1.18 DataStatus ArcDMCSRMSRMClient::rename (SRMClientRequest & *req*, const URL & *newurl*) [inline, virtual]

Rename the URL in *req* to *newurl*. *req* The request object

Parameters:

newurl The new URL

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

5.79.1.19 DataStatus ArcDMCSRMSRMClient::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:

req The request object

Returns:

DataStatus specifying outcome of operation

Implements [ArcDMCSRMSRMClient](#).

5.79.1.20 DataStatus ArcDMCSRMSRMClient::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:

req The request object to query the status of

Returns:

DataStatus specifying outcome of operation

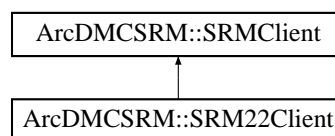
Implements [ArcDMCSRMSRMClient](#).

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

- SRM1Client.h

5.80 ArcDMCSRMSRM22Client Class Reference

Inheritance diagram for ArcDMCSRMSRM22Client::



Public Member Functions

- [SRM22Client](#) (const UserConfig &usercfg, const [SRMURL](#) &url)
- [~SRM22Client](#) ()
- DataStatus [ping](#) (std::string &version)
- DataStatus [getSpaceTokens](#) (std::list< std::string > &tokens, const std::string &description="")
- DataStatus [getRequestTokens](#) (std::list< std::string > &tokens, const std::string &description="")
- DataStatus [getURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- DataStatus [getURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- DataStatus [putURLs](#) (SRMClientRequest &req, std::list< std::string > &urls)
- DataStatus [putURLsStatus](#) (SRMClientRequest &req, std::list< std::string > &urls)
- DataStatus [requestBringOnline](#) (SRMClientRequest &req)
- DataStatus [requestBringOnlineStatus](#) (SRMClientRequest &req)
- DataStatus [info](#) (SRMClientRequest &req, std::map< std::string, std::list< struct [SRMFileMetaData](#) > > &metadata)
- DataStatus [info](#) (SRMClientRequest &req, std::list< struct [SRMFileMetaData](#) > &metadata)
- DataStatus [releaseGet](#) (SRMClientRequest &req)
- DataStatus [releasePut](#) (SRMClientRequest &req)
- DataStatus [release](#) (SRMClientRequest &, bool)
- DataStatus [abort](#) (SRMClientRequest &req, bool source)
- DataStatus [remove](#) (SRMClientRequest &req)
- DataStatus [copy](#) (SRMClientRequest &req, const std::string &source)
- DataStatus [mkDir](#) (SRMClientRequest &req)
- DataStatus [rename](#) (SRMClientRequest &req, const URL &newurl)
- DataStatus [checkPermissions](#) (SRMClientRequest &req)

5.80.1 Constructor & Destructor Documentation

5.80.1.1 ArcDMCSRMSRM22Client::SRM22Client (const UserConfig & usercfg, const SRMURL & url)

Constructor

5.80.1.2 ArcDMCSRMSRM22Client::~~SRM22Client ()

Destructor

5.80.2 Member Function Documentation

5.80.2.1 `DataStatus ArcDMCSRMSRM22Client::abort (SRMClientRequest & req, bool source)` `[virtual]`

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

Implements [ArcDMCSRMSRMClient](#).

5.80.2.2 `DataStatus ArcDMCSRMSRM22Client::checkPermissions (SRMClientRequest & req)` `[virtual]`

Call srmCheckPermission

Implements [ArcDMCSRMSRMClient](#).

5.80.2.3 `DataStatus ArcDMCSRMSRM22Client::copy (SRMClientRequest & req, const std::string & source)` `[virtual]`

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

Implements [ArcDMCSRMSRMClient](#).

5.80.2.4 `DataStatus ArcDMCSRMSRM22Client::getRequestTokens (std::list< std::string > & tokens, const std::string & description = "")` `[virtual]`

Use srmGetRequestTokens to return a list of spaces available

Implements [ArcDMCSRMSRMClient](#).

5.80.2.5 `DataStatus ArcDMCSRMSRM22Client::getSpaceTokens (std::list< std::string > & tokens, const std::string & description = "")` `[virtual]`

Use srmGetSpaceTokens to return a list of spaces available

Implements [ArcDMCSRMSRMClient](#).

5.80.2.6 `DataStatus ArcDMCSRMSRM22Client::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 [ArcDMCSRMSRMClient](#).

5.80.2.7 `DataStatus ArcDMCSRMSRM22Client::getTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls)` `[virtual]`

Uses srmStatusOfGetRequest to query the status of the given request.

Implements [ArcDMCSRMSRMClient](#).

5.80.2.8 DataStatus ArcDMCSRMSRM22Client::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 [ArcDMCSRMSRMClient](#).

5.80.2.9 DataStatus ArcDMCSRMSRM22Client::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 [ArcDMCSRMSRMClient](#).

5.80.2.10 DataStatus ArcDMCSRMSRM22Client::mkDir (SRMClientRequest & req) [virtual]

Call srmMkDir

Implements [ArcDMCSRMSRMClient](#).

5.80.2.11 DataStatus ArcDMCSRMSRM22Client::ping (std::string & version) [virtual]

Get the server version from srmPing

Implements [ArcDMCSRMSRMClient](#).

5.80.2.12 DataStatus ArcDMCSRMSRM22Client::putTURLs (SRMClientRequest & req, std::list< std::string > & urls) [virtual]

Retrieve TURLs which a file can be written to. Uses srmPrepareToPut and waits until a suitable TURL has been assigned if the request is synchronous. If not it returns after making the request. Although a list is returned, SRMV2.2 only returns one TURL per SURL.

Implements [ArcDMCSRMSRMClient](#).

5.80.2.13 DataStatus ArcDMCSRMSRM22Client::putTURLsStatus (SRMClientRequest & req, std::list< std::string > & urls) [virtual]

Uses srmStatusOfPutRequest to query the status of the given request.

Implements [ArcDMCSRMSRMClient](#).

5.80.2.14 DataStatus ArcDMCSRMSRM22Client::release (SRMClientRequest &, bool) [inline, virtual]

Not used in this version of SRM

Implements [ArcDMCSRMSRMClient](#).

5.80.2.15 DataStatus ArcDMCSRMSRM22Client::releaseGet (SRMClientRequest & req) [virtual]

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

Implements [ArcDMCSRMSRMClient](#).

5.80.2.16 DataStatus ArcDMCSRMSRM22Client::releasePut (SRMClientRequest & req) [virtual]

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

Implements [ArcDMCSRMSRMClient](#).

5.80.2.17 DataStatus ArcDMCSRMSRM22Client::remove (SRMClientRequest & req) [virtual]

Delete by srmRm or srmRmdir

Implements [ArcDMCSRMSRMClient](#).

5.80.2.18 DataStatus ArcDMCSRMSRM22Client::rename (SRMClientRequest & req, const URL & newurl) [virtual]

Call srmMv

Implements [ArcDMCSRMSRMClient](#).

5.80.2.19 DataStatus ArcDMCSRMSRM22Client::requestBringOnline (SRMClientRequest & req) [virtual]

Call srmBringOnline with the URLs specified in req.

Implements [ArcDMCSRMSRMClient](#).

5.80.2.20 DataStatus ArcDMCSRMSRM22Client::requestBringOnlineStatus (SRMClientRequest & req) [virtual]

Call srmStatusOfBringOnlineRequest and update req with any changes.

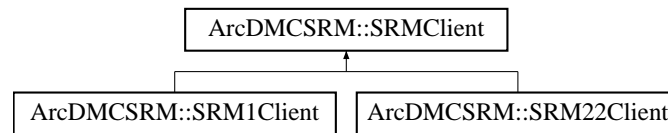
Implements [ArcDMCSRMSRMClient](#).

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

- SRM22Client.h

5.81 ArcDMCSRMSRMClient Class Reference

#include <SRMClient.h> Inheritance diagram for ArcDMCSRMSRMClient:



Public Member Functions

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

Static Public Member Functions

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

Protected Member Functions

- [SRMClient](#) (const UserConfig &usercfg, const [SRMURL](#) &url)
- DataStatus [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](#)

5.81.1 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.

5.81.2 Constructor & Destructor Documentation

5.81.2.1 virtual ArcDMCSRMSRMClient::~SRMClient () [virtual]

Destructor

5.81.3 Member Function Documentation

5.81.3.1 virtual DataStatus ArcDMCSRMSRMClient::abort (SRMClientRequest & *req*, bool *source*) [pure virtual]

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

Parameters:

- req* The request object
- source* Whether source or destination is being aborted

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.2 virtual DataStatus ArcDMCSRMSRMClient::checkPermissions (SRMClientRequest & *req*) [pure virtual]

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

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.3 virtual DataStatus ArcDMCSRMSRMClient::copy (SRMClientRequest & req, const std::string & source) [pure virtual]

Copy a file between two SRM storages.

Parameters:

req The request object

source The source SURL

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.4 static SRMClient* ArcDMCSRMSRMClient::getInstance (const UserConfig & usercfg, const std::string & url, std::string & error) [static]

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

Parameters:

usercfg The user configuration.

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

error Details of error if one occurred

Returns:

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

5.81.3.5 virtual DataStatus ArcDMCSRMSRMClient::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. Was used by the old ngstage command but is currently unused.

Parameters:

tokens The list filled by the service

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

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.6 **virtual DataStatus ArcDMCSRMSRMClient::getSpaceTokens (std::list< std::string > & tokens, const std::string & description = "") [pure virtual]**

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

Parameters:

tokens The list filled by the service
description The space token description

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.7 **virtual DataStatus ArcDMCSRMSRMClient::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:

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

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.8 **virtual DataStatus ArcDMCSRMSRMClient::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:

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

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.9 `std::string ArcDMCSRMSRMClient::getVersion () const [inline]`

Returns the version of the SRM protocol used by this instance

References version.

5.81.3.10 `virtual DataStatus ArcDMCSRMSRMClient::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 ≥ 1 then the list in metadata will contain the content of the directory.

Parameters:

req The request object

metadata A list of structs filled with file information.

Returns:

DataStatus specifying outcome of operation

See also:

[SRMFileMetaData](#)

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.11 `virtual DataStatus ArcDMCSRMSRMClient::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 ≥ 1 then the list mapped to each SURL in metadata will contain the content of the directory or directories.

Parameters:

req The request object

metadata 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:

DataStatus specifying outcome of operation

See also:

[SRMFileMetaData](#)

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.12 `virtual DataStatus ArcDMCSRMSRMClient::mkDir (SRMClientRequest & req) [pure virtual]`

Make required directories for the SURL in the request

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.13 virtual DataStatus ArcDMCSRMSRMClient::ping (std::string & version) [pure virtual]

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

Parameters:

version The version returned by the server

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.14 virtual DataStatus ArcDMCSRMSRMClient::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:

req The request object

urls A list of TURLs filled by the method

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.15 virtual DataStatus ArcDMCSRMSRMClient::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:

req The request object. Status must be ongoing.

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

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.16 virtual DataStatus ArcDMCSRMSRMClient::release (SRMClientRequest & req, bool source) [pure virtual]

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

Parameters:

req The request object

source Whether source or destination is being released

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.17 virtual DataStatus ArcDMCSRMSRMClient::releaseGet (SRMClientRequest & req) [pure virtual]

Should be called after a successful copy from SRM storage.

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.18 virtual DataStatus ArcDMCSRMSRMClient::releasePut (SRMClientRequest & req) [pure virtual]

Should be called after a successful copy to SRM storage.

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.19 **virtual DataStatus ArcDMCSRMSRMClient::remove (SRMClientRequest & req) [pure virtual]**

Delete a file physically from storage and the SRM namespace.

Parameters:

req The request object

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.20 **virtual DataStatus ArcDMCSRMSRMClient::rename (SRMClientRequest & req, const URL & newurl) [pure virtual]**

Rename the URL in req to newurl. *req* The request object

Parameters:

newurl The new URL

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.21 **virtual DataStatus ArcDMCSRMSRMClient::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:

req The request object

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.3.22 **virtual DataStatus ArcDMCSRMSRMClient::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:

req The request object to query the status of

Returns:

DataStatus specifying outcome of operation

Implemented in [ArcDMCSRMSRM1Client](#), and [ArcDMCSRMSRM22Client](#).

5.81.4 Field Documentation

5.81.4.1 std::string ArcDMCSRMSRMClient::service_endpoint [protected]

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

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

- SRMClient.h

5.82 ArcDMCSRMSRMClientRequest Class Reference

Class to represent a SRM request.

```
#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

5.82.1 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 releaseGet().

5.82.2 Constructor & Destructor Documentation

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

Creates a request object with multiple URLs. The URLs here are in the form srm://srm.host.org/path/to/file

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

Creates a request object with a single URL. The URL here is in the form srm://srm.host.org/path/to/file

5.82.3 Member Function Documentation

5.82.3.1 int ArcDMCSRMSRMClientRequest::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

5.83 ArcDMCSRMSRMFileInfo Class Reference

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

5.83.1 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

5.84 ArcDMCSR::SRMFileMetaData Struct Reference

SRM-related file metadata.

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

5.84.1 Detailed Description

SRM-related file metadata.

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

- SRMClient.h

5.85 ArcDMCSRm::SRMInfo Class Reference

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

5.85.1 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

5.86 ArcDMCSRMSRMInvalidRequestException Class Reference

General exception to represent a bad SRM request.

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

5.86.1 Detailed Description

General exception to represent a bad SRM request.

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

- SRMClientRequest.h

5.87 ArcDMCSRML::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](#) ()

5.87.1 Constructor & Destructor Documentation

5.87.1.1 ArcDMCSRML::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

5.87.2 Member Function Documentation

5.87.2.1 std::string ArcDMCSRML::SRMURL::BaseURL (void) const

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

5.87.2.2 std::string ArcDMCSRML::SRMURL::ContactURL (void) const

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

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

eg /srm/managerv2

5.87.2.4 std::string ArcDMCSRML::SRMURL::FileName (void) const [inline]

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

5.87.2.5 std::string ArcDMCSRML::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

5.87.2.6 bool ArcDMCSRML::SRMURL::PortDefined () [inline]

Was the port number given in the constructor?

5.87.2.7 void ArcDMCSRMSRMURL::SetSRMVersion (const std::string & *version*)

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

5.87.2.8 std::string ArcDMCSRMSRMURL::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

5.88 ArcDMCHTTP::StreamBuffer Class Reference

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

- StreamBuffer.h

5.89 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

5.90 ArcSec::UsernameTokenSH Class Reference

Adds WS-Security Username Token into SOAP Header.

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

5.90.1 Detailed Description

Adds WS-Security Username Token into SOAP Header.

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

- UsernameTokenSH.h

5.91 ArcSHCLegacy::voms Struct Reference

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

Data Fields

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

5.91.1 Detailed Description

VOMS data

5.91.2 Field Documentation

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

User's characteristics

5.91.2.2 std::string ArcSHCLegacy::voms::server

The VOMS server DN, as from its certificate

5.91.2.3 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](#)

5.92 ArcSHCLegacy::voms_attrs Struct Reference

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

Data Fields

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

5.92.1 Detailed Description

VOMS attributes

5.92.2 Field Documentation

5.92.2.1 std::string ArcSHCLegacy::voms_attrs::cap

user's capability

5.92.2.2 std::string ArcSHCLegacy::voms_attrs::group

user's group

5.92.2.3 std::string ArcSHCLegacy::voms_attrs::role

user's role

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

- [auth.h](#)

5.93 ArcSec::X509TokenSH Class Reference

Adds WS-Security X509 Token into SOAP Header.

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

5.93.1 Detailed Description

Adds WS-Security X509 Token into SOAP Header.

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

- X509TokenSH.h

5.94 ArcSec::XACMLAlgFactory Class Reference

Algorithm factory class for XACML.

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

Public Member Functions

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

5.94.1 Detailed Description

Algorithm factory class for XACML.

5.94.2 Member Function Documentation

5.94.2.1 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

5.95 ArcSec::XACMLApply Class Reference

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

- XACMLApply.h

5.96 ArcSec::XACMLAttributeFactory Class Reference

Attribute factory class for XACML specified attributes.

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

Public Member Functions

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

5.96.1 Detailed Description

Attribute factory class for XACML specified attributes.

5.96.2 Member Function Documentation

5.96.2.1 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

5.97 ArcSec::XACMLAttributeProxy< TheAttribute > Class Template Reference

XACML specific AttributeProxy class.

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

Public Member Functions

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

5.97.1 Detailed Description

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

XACML specific AttributeProxy class.

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

- XACMLAttributeProxy.h

5.98 ArcSec::XACMLCondition Class Reference

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

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

Public Member Functions

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

5.98.1 Detailed Description

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

5.98.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLCondition.h

5.99 ArcSec::XACMLEvaluationCtx Class Reference

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

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

Public Member Functions

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

5.99.1 Detailed Description

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

5.99.2 Constructor & Destructor Documentation

5.99.2.1 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

5.100 ArcSec::XACMLEvaluator Class Reference

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

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

Public Member Functions

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

5.100.1 Detailed Description

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

5.100.2 Member Function Documentation

5.100.2.1 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

5.101 ArcSec::XACMLFnFactory Class Reference

Function factory class for XACML specified attributes.

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

Public Member Functions

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

5.101.1 Detailed Description

Function factory class for XACML specified attributes.

5.101.2 Member Function Documentation

5.101.2.1 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

5.102 ArcSec::XACMLPDP Class Reference

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

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

5.102.1 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

5.103 ArcSec::XACMLPolicy Class Reference

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

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

Public Member Functions

- [XACMLPolicy](#) (Arc::PluginArgument *parg)
- [XACMLPolicy](#) (const Arc::XMLNode node, Arc::PluginArgument *parg)
- [XACMLPolicy](#) (const Arc::XMLNode node, EvaluatorContext *ctx, Arc::PluginArgument *parg)
- virtual void [make_policy](#) ()

5.103.1 Detailed Description

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

5.103.2 Constructor & Destructor Documentation

5.103.2.1 ArcSec::XACMLPolicy::XACMLPolicy (Arc::PluginArgument *parg)

Constructor

5.103.2.2 ArcSec::XACMLPolicy::XACMLPolicy (const Arc::XMLNode node, Arc::PluginArgument *parg)

Constructor

5.103.2.3 ArcSec::XACMLPolicy::XACMLPolicy (const Arc::XMLNode node, EvaluatorContext *ctx, Arc::PluginArgument *parg)

Constructor -

5.103.3 Member Function Documentation

5.103.3.1 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

5.104 ArcSec::XACMLRequest Class Reference

Public Member Functions

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

5.104.1 Member Function Documentation

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

Get the name of corresponding evaluator

5.104.1.2 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

5.105 ArcSec::XACMLRule Class Reference

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

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

5.105.1 Detailed Description

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

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

- XACMLRule.h

5.106 ArcSec::XACMLTarget Class Reference

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

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

Public Member Functions

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

5.106.1 Detailed Description

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

5.106.2 Constructor & Destructor Documentation

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

Constructor -

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

- XACMLTarget.h

5.107 ArcSec::XACMLTargetMatch Class Reference

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

- XACMLTarget.h

5.108 ArcSec::XACMLTargetMatchGroup Class Reference

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

- XACMLTarget.h

5.109 ArcSec::XACMLTargetSection Class Reference

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

- XACMLTarget.h

Index

- ~LDAPQuery
 - ArcDMCLDAP::LDAPQuery, [57](#)
- ~PayloadTLSSStream
 - ArcMCCTLS::PayloadTLSSStream, [93](#)
- ~SRM22Client
 - ArcDMCSRMSRM::SRM22Client, [107](#)
- ~SRMClient
 - ArcDMCSRMSRM::SRMClient, [112](#)
- abort
 - ArcDMCSRMSRM::SRM1Client, [100](#)
 - ArcDMCSRMSRM::SRM22Client, [108](#)
 - ArcDMCSRMSRM::SRMClient, [112](#)
- AndList
 - ArcSec, [13](#)
- ArcDMCARC::DataPointARC, [35](#)
- ArcDMCFile::DataPointFile, [36](#)
- ArcDMCGFAL::DataPointGFAL, [37](#)
- ArcDMCGFAL::GFALTransfer3rdParty, [55](#)
- ArcDMCGFAL::GFALUtils, [56](#)
- ArcDMCGridFTP::DataPointGridFTP, [38](#)
- ArcDMCGridFTP::Lister, [63](#)
- ArcDMCHTTP::DataPointHTTP, [39](#)
- ArcDMCHTTP::StreamBuffer, [127](#)
- ArcDMCLDAP::DataPointLDAP, [40](#)
- ArcDMCLDAP::LDAPQuery, [57](#)
 - ~LDAPQuery, [57](#)
 - LDAPQuery, [57](#)
 - Query, [57](#)
 - Result, [57](#)
- ArcDMCLFC::DataPointLFC, [41](#)
- ArcDMCMock::DataPointMock, [42](#)
- ArcDMCSRMSRM::DataPointSRM, [43](#)
- ArcDMCSRMSRM::SRM1Client, [100](#)
 - abort, [100](#)
 - checkPermissions, [100](#)
 - copy, [101](#)
 - getRequestTokens, [101](#)
 - getSpaceTokens, [101](#)
 - getURLs, [102](#)
 - getURLsStatus, [102](#)
 - info, [102](#), [103](#)
 - mkDir, [103](#)
 - ping, [103](#)
 - putURLs, [104](#)
 - putURLsStatus, [104](#)
 - release, [104](#)
 - releaseGet, [105](#)
 - releasePut, [105](#)
 - remove, [105](#)
 - rename, [106](#)
 - requestBringOnline, [106](#)
 - requestBringOnlineStatus, [106](#)
- ArcDMCSRMSRM::SRM22Client, [107](#)
 - ~SRM22Client, [107](#)
 - abort, [108](#)
 - checkPermissions, [108](#)
 - copy, [108](#)
 - getRequestTokens, [108](#)
 - getSpaceTokens, [108](#)
 - getURLs, [108](#)
 - getURLsStatus, [108](#)
 - info, [108](#), [109](#)
 - mkDir, [109](#)
 - ping, [109](#)
 - putURLs, [109](#)
 - putURLsStatus, [109](#)
 - release, [109](#)
 - releaseGet, [109](#)
 - releasePut, [110](#)
 - remove, [110](#)
 - rename, [110](#)
 - requestBringOnline, [110](#)
 - requestBringOnlineStatus, [110](#)
 - SRM22Client, [107](#)
- ArcDMCSRMSRM::SRMClient, [111](#)
 - ~SRMClient, [112](#)
 - abort, [112](#)
 - checkPermissions, [112](#)
 - copy, [113](#)
 - getInstance, [113](#)
 - getRequestTokens, [113](#)
 - getSpaceTokens, [113](#)
 - getURLs, [114](#)
 - getURLsStatus, [114](#)
 - getVersion, [114](#)
 - info, [115](#)
 - mkDir, [115](#)
 - ping, [116](#)
 - putURLs, [116](#)

- putURLsStatus, 116
- release, 117
- releaseGet, 117
- releasePut, 117
- remove, 117
- rename, 118
- requestBringOnline, 118
- requestBringOnlineStatus, 118
- service_endpoint, 119
- ArcDMCSRMR::SRMClientRequest, 120
 - SRMClientRequest, 120
 - waiting_time, 120
- ArcDMCSRMR::SRMFileInfo, 121
- ArcDMCSRMR::SRMFileMetaData, 122
- ArcDMCSRMR::SRMInfo, 123
- ArcDMCSRMR::SRMInvalidRequestException, 124
- ArcDMCSRMR::SRMURL, 125
 - BaseURL, 125
 - ContactURL, 125
 - Endpoint, 125
 - FileName, 125
 - FullURL, 125
 - PortDefined, 125
 - SetSRMVersion, 125
 - ShortURL, 126
 - SRMURL, 125
- ArcDMCXrootd::DataPointXrootd, 44
- ArcEvaluationCtx
 - ArcSec::ArcEvaluationCtx, 20
- ArcMCCGSI::MCC_GSI_Client, 64
- ArcMCCGSI::MCC_GSI_Service, 65
- ArcMCCGSI::PayloadGSISStream, 80
- ArcMCCHTTP::MCC_HTTP, 66
- ArcMCCHTTP::MCC_HTTP_Client, 67
- ArcMCCHTTP::MCC_HTTP_Service, 68
- ArcMCCHTTP::PayloadHTTP, 81
 - Attribute, 82
 - Attributes, 82
 - attributes_, 82
 - code_, 82
 - end_, 82
 - GetError, 82
 - keep_alive_, 82
 - length_, 82
 - method_, 82
 - offset_, 83
 - PayloadHTTP, 81, 82
 - reason_, 83
 - size_, 83
 - uri_, 83
 - version_major_, 83
 - version_minor_, 83
- ArcMCCHTTP::PayloadHTTPIn, 84
 - body_, 85
 - chunk_size_, 85
 - fetched_, 85
 - get_body, 84
 - PayloadHTTPIn, 84
 - read, 84
 - read_header, 85
 - readline, 85
 - stream_offset_, 85
 - stream_own_, 85
 - tbuf_, 85
 - tbuflen_, 85
- ArcMCCHTTP::PayloadHTTPOut, 86
 - Attribute, 87
 - Flush, 87
 - header_, 87
 - make_header, 87
 - PayloadHTTPOut, 86
 - rbody_, 87
 - sbody_, 87
 - sbody_size_, 87
 - stream_offset_, 87
 - to_stream_, 87
 - use_chunked_transfer_, 87
- ArcMCCHTTP::PayloadHTTPOutRaw, 89
 - Body, 89
- ArcMCCHTTP::PayloadHTTPOutStream, 90
 - Body, 90
- ArcMCCMsgValidator::MCC_MsgValidator, 69
- ArcMCCMsgValidator::MCC_MsgValidator_Service, 70
- ArcMCCSOAP::MCC_SOAP, 71
- ArcMCCSOAP::MCC_SOAP_Client, 72
- ArcMCCSOAP::MCC_SOAP_Service, 73
- ArcMCCTCP::MCC_TCP, 74
- ArcMCCTCP::MCC_TCP_Client, 75
- ArcMCCTCP::MCC_TCP_Service, 76
 - MCC_TCP_Service, 76
- ArcMCCTCP::PayloadTCPSocket, 91
 - PayloadTCPSocket, 91
- ArcMCCTLS::ConfigTLSMCC, 34
- ArcMCCTLS::MCC_TLS, 77
- ArcMCCTLS::MCC_TLS_Client, 78
- ArcMCCTLS::MCC_TLS_Service, 79
- ArcMCCTLS::PayloadTLSMCC, 92
 - PayloadTLSMCC, 92
- ArcMCCTLS::PayloadTLSSStream, 93
 - ~PayloadTLSSStream, 93
 - GetCert, 93
 - GetPeerCert, 93
 - PayloadTLSSStream, 93
 - ssl_, 94
 - STACK_OF, 94
- ArcMCCTLSec::DelegationCollector, 45
- ArcMCCTLSec::DelegationMultiSecAttr, 46

- ArcMCCTLSec::DelegationSecAttr, 48
- ArcPolicy
 - ArcSec::ArcPolicy, 24
- ArcSec, 11
 - AndList, 13
 - Match, 14
- ArcSec::AllowPDP, 15
- ArcSec::ArcAlgFactory, 16
 - createAlg, 16
- ArcSec::ArcAttributeFactory, 17
 - createValue, 17
- ArcSec::ArcAttributeProxy, 18
- ArcSec::ArcAuthZ, 19
 - Handle, 19
 - MakePDPs, 19
- ArcSec::ArcEvaluationCtx, 20
 - ArcEvaluationCtx, 20
 - split, 20
- ArcSec::ArcEvaluator, 21
 - evaluate, 21
- ArcSec::ArcFnFactory, 22
 - createFn, 22
- ArcSec::ArcPDP, 23
- ArcSec::ArcPolicy, 24
 - ArcPolicy, 24
 - make_policy, 24
- ArcSec::ArcRequest, 25
- ArcSec::ArcRequestItem, 26
- ArcSec::ArcRequestTuple, 27
- ArcSec::ArcRule, 28
- ArcSec::AttributeDesignator, 29
- ArcSec::AttributeSelector, 30
- ArcSec::DelegationPDP, 47
- ArcSec::DelegationSH, 49
- ArcSec::DenyPDP, 50
- ArcSec::GACLEvaluator, 51
 - evaluate, 51
- ArcSec::GACLPDP, 52
- ArcSec::GACLPolicy, 53
- ArcSec::GACLRequest, 54
- ArcSec::PDPSERVICEInvoker, 95
- ArcSec::SAML2SSO_AssertionConsumerSH, 96
- ArcSec::SAMLTokenSH, 97
- ArcSec::SimpleListPDP, 98
- ArcSec::UsernameTokenSH, 129
- ArcSec::X509TokenSH, 132
- ArcSec::XACMLAlgFactory, 133
 - createAlg, 133
- ArcSec::XACMLApply, 134
- ArcSec::XACMLAttributeFactory, 135
 - createValue, 135
- ArcSec::XACMLAttributeProxy, 136
- ArcSec::XACMLCondition, 137
 - XACMLCondition, 137
- ArcSec::XACMLEvaluationCtx, 138
 - XACMLEvaluationCtx, 138
- ArcSec::XACMLEvaluator, 139
 - evaluate, 139
- ArcSec::XACMLFnFactory, 140
 - createFn, 140
- ArcSec::XACMLPDP, 141
- ArcSec::XACMLPolicy, 142
 - make_policy, 142
 - XACMLPolicy, 142
- ArcSec::XACMLRequest, 143
 - getEvalName, 143
 - getName, 143
- ArcSec::XACMLRule, 144
- ArcSec::XACMLTarget, 145
 - XACMLTarget, 145
- ArcSec::XACMLTargetMatch, 146
- ArcSec::XACMLTargetMatchGroup, 147
- ArcSec::XACMLTargetSection, 148
- ArcSHCLegacy::AuthUser, 31
- ArcSHCLegacy::AuthVO, 32
- ArcSHCLegacy::ConfigParser, 33
- ArcSHCLegacy::LegacyMap, 59
- ArcSHCLegacy::LegacyPDP, 60
- ArcSHCLegacy::LegacySecAttr, 61
- ArcSHCLegacy::LegacySecHandler, 62
- ArcSHCLegacy::SimpleMap, 99
- ArcSHCLegacy::UnixMap, 128
- ArcSHCLegacy::voms, 130
 - attrs, 130
 - server, 130
 - voname, 130
- ArcSHCLegacy::voms_attrs, 131
 - cap, 131
 - group, 131
 - role, 131
- Attribute
 - ArcMCCHTTP::PayloadHTTP, 82
 - ArcMCCHTTP::PayloadHTTPOut, 87
- Attributes
 - ArcMCCHTTP::PayloadHTTP, 82
- attributes_
 - ArcMCCHTTP::PayloadHTTP, 82
- attrs
 - ArcSHCLegacy::voms, 130
- BaseURL
 - ArcDMCSR::SRMURL, 125
- Body
 - ArcMCCHTTP::PayloadHTTPOutRaw, 89
 - ArcMCCHTTP::PayloadHTTPOutStream, 90
- body_
 - ArcMCCHTTP::PayloadHTTPIn, 85

- cap
 - ArcSHCLegacy::voms_attrs, 131
- checkPermissions
 - ArcDMCSRMSRM1Client, 100
 - ArcDMCSRMSRM22Client, 108
 - ArcDMCSRMSRMClient, 112
- chunk_size_
 - ArcMCCHTTP::PayloadHTTPIn, 85
- code_
 - ArcMCCHTTP::PayloadHTTP, 82
- ContactURL
 - ArcDMCSRMSRMURL, 125
- copy
 - ArcDMCSRMSRM1Client, 101
 - ArcDMCSRMSRM22Client, 108
 - ArcDMCSRMSRMClient, 113
- createAlg
 - ArcSec::ArcAlgFactory, 16
 - ArcSec::XACMLAlgFactory, 133
- createFn
 - ArcSec::ArcFnFactory, 22
 - ArcSec::XACMLFnFactory, 140
- createValue
 - ArcSec::ArcAttributeFactory, 17
 - ArcSec::XACMLAttributeFactory, 135
- end_
 - ArcMCCHTTP::PayloadHTTP, 82
- Endpoint
 - ArcDMCSRMSRMURL, 125
- evaluate
 - ArcSec::ArcEvaluator, 21
 - ArcSec::GACLEvaluator, 51
 - ArcSec::XACMLEvaluator, 139
- fetched_
 - ArcMCCHTTP::PayloadHTTPIn, 85
- FileName
 - ArcDMCSRMSRMURL, 125
- Flush
 - ArcMCCHTTP::PayloadHTTPOut, 87
- FullURL
 - ArcDMCSRMSRMURL, 125
- get_body
 - ArcMCCHTTP::PayloadHTTPIn, 84
- GetCert
 - ArcMCCTLS::PayloadTLSSStream, 93
- GetError
 - ArcMCCHTTP::PayloadHTTP, 82
- getEvalName
 - ArcSec::XACMLRequest, 143
- getInstance
 - ArcDMCSRMSRMClient, 113
- getName
 - ArcSec::XACMLRequest, 143
- GetPeerCert
 - ArcMCCTLS::PayloadTLSSStream, 93
- getRequestTokens
 - ArcDMCSRMSRM1Client, 101
 - ArcDMCSRMSRM22Client, 108
 - ArcDMCSRMSRMClient, 113
- getSpaceTokens
 - ArcDMCSRMSRM1Client, 101
 - ArcDMCSRMSRM22Client, 108
 - ArcDMCSRMSRMClient, 113
- getTURLs
 - ArcDMCSRMSRM1Client, 102
 - ArcDMCSRMSRM22Client, 108
 - ArcDMCSRMSRMClient, 114
- getTURLsStatus
 - ArcDMCSRMSRM1Client, 102
 - ArcDMCSRMSRM22Client, 108
 - ArcDMCSRMSRMClient, 114
- getVersion
 - ArcDMCSRMSRMClient, 114
- group
 - ArcSHCLegacy::voms_attrs, 131
- Handle
 - ArcSec::ArcAuthZ, 19
- header_
 - ArcMCCHTTP::PayloadHTTPOut, 87
- info
 - ArcDMCSRMSRM1Client, 102, 103
 - ArcDMCSRMSRM22Client, 108, 109
 - ArcDMCSRMSRMClient, 115
- keep_alive_
 - ArcMCCHTTP::PayloadHTTP, 82
- LDAPQuery
 - ArcDMCLDAP::LDAPQuery, 57
- length_
 - ArcMCCHTTP::PayloadHTTP, 82
- make_header
 - ArcMCCHTTP::PayloadHTTPOut, 87
- make_policy
 - ArcSec::ArcPolicy, 24
 - ArcSec::XACMLPolicy, 142
- MakePDPs
 - ArcSec::ArcAuthZ, 19
- Match
 - ArcSec, 14
- MCC_TCP_Service
 - ArcMCCTCP::MCC_TCP_Service, 76
- method_

- ArcMCCHTTP::PayloadHTTP, 82
- mkDir
 - ArcDMCSRMSRM1Client, 103
 - ArcDMCSRMSRM22Client, 109
 - ArcDMCSRMSRMClient, 115
- offset_
 - ArcMCCHTTP::PayloadHTTP, 83
- PayloadHTTP
 - ArcMCCHTTP::PayloadHTTP, 81, 82
- PayloadHTTPIn
 - ArcMCCHTTP::PayloadHTTPIn, 84
- PayloadHTTPOut
 - ArcMCCHTTP::PayloadHTTPOut, 86
- PayloadTCPsocket
 - ArcMCCTCP::PayloadTCPsocket, 91
- PayloadTSLMCC
 - ArcMCCTLS::PayloadTSLMCC, 92
- PayloadTLSSStream
 - ArcMCCTLS::PayloadTLSSStream, 93
- ping
 - ArcDMCSRMSRM1Client, 103
 - ArcDMCSRMSRM22Client, 109
 - ArcDMCSRMSRMClient, 116
- PortDefined
 - ArcDMCSRMSRMURL, 125
- putURLs
 - ArcDMCSRMSRM1Client, 104
 - ArcDMCSRMSRM22Client, 109
 - ArcDMCSRMSRMClient, 116
- putURLsStatus
 - ArcDMCSRMSRM1Client, 104
 - ArcDMCSRMSRM22Client, 109
 - ArcDMCSRMSRMClient, 116
- Query
 - ArcDMCLDAP::LDAPQuery, 57
- rbody_
 - ArcMCCHTTP::PayloadHTTPOut, 87
- read
 - ArcMCCHTTP::PayloadHTTPIn, 84
- read_header
 - ArcMCCHTTP::PayloadHTTPIn, 85
- readline
 - ArcMCCHTTP::PayloadHTTPIn, 85
- reason_
 - ArcMCCHTTP::PayloadHTTP, 83
- release
 - ArcDMCSRMSRM1Client, 104
 - ArcDMCSRMSRM22Client, 109
 - ArcDMCSRMSRMClient, 117
- releaseGet
 - ArcDMCSRMSRM1Client, 105
 - ArcDMCSRMSRM22Client, 109
 - ArcDMCSRMSRMClient, 117
- releasePut
 - ArcDMCSRMSRM1Client, 105
 - ArcDMCSRMSRM22Client, 110
 - ArcDMCSRMSRMClient, 117
- remove
 - ArcDMCSRMSRM1Client, 105
 - ArcDMCSRMSRM22Client, 110
 - ArcDMCSRMSRMClient, 117
- rename
 - ArcDMCSRMSRM1Client, 106
 - ArcDMCSRMSRM22Client, 110
 - ArcDMCSRMSRMClient, 118
- requestBringOnline
 - ArcDMCSRMSRM1Client, 106
 - ArcDMCSRMSRM22Client, 110
 - ArcDMCSRMSRMClient, 118
- requestBringOnlineStatus
 - ArcDMCSRMSRM1Client, 106
 - ArcDMCSRMSRM22Client, 110
 - ArcDMCSRMSRMClient, 118
- Result
 - ArcDMCLDAP::LDAPQuery, 57
- role
 - ArcSHCLegacy::voms_attrs, 131
- sbody_
 - ArcMCCHTTP::PayloadHTTPOut, 87
- sbody_size_
 - ArcMCCHTTP::PayloadHTTPOut, 87
- server
 - ArcSHCLegacy::voms, 130
- service_endpoint
 - ArcDMCSRMSRMClient, 119
- SetSRMVersion
 - ArcDMCSRMSRMURL, 125
- ShortURL
 - ArcDMCSRMSRMURL, 126
- size_
 - ArcMCCHTTP::PayloadHTTP, 83
- split
 - ArcSec::ArcEvaluationCtx, 20
- SRM22Client
 - ArcDMCSRMSRM22Client, 107
- SRMClientRequest
 - ArcDMCSRMSRMClientRequest, 120
- SRMURL
 - ArcDMCSRMSRMURL, 125
- ssl_
 - ArcMCCTLS::PayloadTLSSStream, 94
- STACK_OF
 - ArcMCCTLS::PayloadTLSSStream, 94

stream_offset_
 ArcMCCHTTP::PayloadHTTPIn, [85](#)
 ArcMCCHTTP::PayloadHTTPOut, [87](#)
stream_own_
 ArcMCCHTTP::PayloadHTTPIn, [85](#)

tbuf_
 ArcMCCHTTP::PayloadHTTPIn, [85](#)
tbuflen_
 ArcMCCHTTP::PayloadHTTPIn, [85](#)
to_stream_
 ArcMCCHTTP::PayloadHTTPOut, [87](#)

uri_
 ArcMCCHTTP::PayloadHTTP, [83](#)
use_chunked_transfer_
 ArcMCCHTTP::PayloadHTTPOut, [87](#)

version_major_
 ArcMCCHTTP::PayloadHTTP, [83](#)
version_minor_
 ArcMCCHTTP::PayloadHTTP, [83](#)
voname
 ArcSHCLegacy::voms, [130](#)

waiting_time
 ArcDMCSRMSRMClientRequest, [120](#)

XACMLCondition
 ArcSec::XACMLCondition, [137](#)
XACMLEvaluationCtx
 ArcSec::XACMLEvaluationCtx, [138](#)
XACMLPolicy
 ArcSec::XACMLPolicy, [142](#)
XACMLTarget
 ArcSec::XACMLTarget, [145](#)