

Hosting Environment (Daemon) Services

Generated by Doxygen 1.7.5

Wed Apr 11 2012 17:13:12

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	DREService Namespace Reference	11
4.1.1	Detailed Description	11
5	Data Structure Documentation	13
5.1	ARex::ARexGMConfig Class Reference	13
5.2	ARex::ARexJob Class Reference	13
5.2.1	Detailed Description	14
5.2.2	Constructor & Destructor Documentation	14
5.2.2.1	ARexJob	14
5.2.2.2	ARexJob	14
5.2.3	Member Function Documentation	14
5.2.3.1	Cancel	14
5.2.3.2	ChooseSessionDir	14
5.2.3.3	Clean	15
5.2.3.4	Created	15
5.2.3.5	CreateFile	15
5.2.3.6	Failed	15

5.2.3.7	FailedState	15
5.2.3.8	Failure	15
5.2.3.9	GetDescription	15
5.2.3.10	ID	15
5.2.3.11	Jobs	15
5.2.3.12	LogDir	15
5.2.3.13	LogFiles	16
5.2.3.14	Modified	16
5.2.3.15	OpenDir	16
5.2.3.16	OpenFile	16
5.2.3.17	OpenLogFile	16
5.2.3.18	Resume	16
5.2.3.19	SessionDir	16
5.2.3.20	State	16
5.2.3.21	State	16
5.2.3.22	TotalJobs	16
5.2.3.23	UpdateCredentials	17
5.3	ARex::ARexService Class Reference	17
5.4	AuthEvaluator Class Reference	17
5.5	AuthUser Class Reference	17
5.6	AuthVO Class Reference	17
5.7	CacheConfig Class Reference	18
5.7.1	Detailed Description	18
5.7.2	Constructor & Destructor Documentation	18
5.7.2.1	CacheConfig	18
5.7.2.2	CacheConfig	18
5.7.3	Member Function Documentation	18
5.7.3.1	parseINIConf	18
5.7.3.2	setCacheDirs	18
5.8	CacheConfigException Class Reference	19
5.8.1	Detailed Description	19
5.9	Cache::CacheService Class Reference	19
5.9.1	Detailed Description	19
5.9.2	Constructor & Destructor Documentation	20

5.9.2.1	CacheService	20
5.9.2.2	~CacheService	20
5.9.3	Member Function Documentation	20
5.9.3.1	CacheCheck	20
5.9.3.2	CacheLink	20
5.9.3.3	operator bool	20
5.9.3.4	operator!	20
5.9.3.5	process	21
5.9.3.6	RegistrationCollector	21
5.10	ArcSec::Charon Class Reference	21
5.10.1	Detailed Description	21
5.11	GridFTP_Commands::close_semaphore_t Class Reference	21
5.12	ContinuationPlugins::command_t Class Reference	21
5.13	CommFIFO Class Reference	22
5.14	gridftpd::ConfigSections Class Reference	22
5.15	ConfigSections Class Reference	22
5.16	ARex::DelegationStore::Consumer Class Reference	22
5.17	ContinuationPlugins Class Reference	22
5.18	ARex::CountedResource Class Reference	23
5.19	gridftpd::Daemon Class Reference	23
5.20	GridFTP_Commands::data_buffer_t Struct Reference	23
5.21	DataStaging::DataDeliveryService Class Reference	23
5.21.1	Detailed Description	23
5.22	ARex::DelegationStore Class Reference	24
5.23	ARex::DelegationStores Class Reference	24
5.24	DirectAccess::diraccess_t Struct Reference	24
5.25	DirectAccess Class Reference	25
5.26	DirectFilePlugin Class Reference	25
5.27	DirEntry Class Reference	25
5.28	DREService::DREWebService Class Reference	25
5.28.1	Constructor & Destructor Documentation	26
5.28.1.1	DREWebService	26
5.28.1.2	~DREWebService	26
5.28.2	Member Function Documentation	26

5.28.2.1	makeFault	26
5.28.2.2	process	26
5.28.3	Field Documentation	27
5.28.3.1	logger	27
5.28.3.2	ns_	27
5.29	DTRGenerator Class Reference	27
5.29.1	Detailed Description	27
5.29.2	Constructor & Destructor Documentation	28
5.29.2.1	DTRGenerator	28
5.29.2.2	~DTRGenerator	28
5.29.3	Member Function Documentation	28
5.29.3.1	cancelJob	28
5.29.3.2	checkUploadedFiles	28
5.29.3.3	hasJob	28
5.29.3.4	queryJobFinished	29
5.29.3.5	receiveDTR	29
5.29.3.6	receiveJob	29
5.29.3.7	removeJob	29
5.30	DTRInfo Class Reference	30
5.30.1	Detailed Description	30
5.30.2	Constructor & Destructor Documentation	30
5.30.2.1	DTRInfo	30
5.31	CommFIFO::elem_t Class Reference	30
5.32	Entry Class Reference	30
5.33	Exec Class Reference	31
5.34	ARex::FileChunks Class Reference	31
5.34.1	Detailed Description	31
5.34.2	Member Function Documentation	31
5.34.2.1	Release	31
5.34.2.2	Remove	31
5.35	ARex::FileChunksList Class Reference	32
5.35.1	Detailed Description	32
5.35.2	Member Function Documentation	32
5.35.2.1	Get	32

5.36	ARex::FileChunksRef Class Reference	32
5.37	FileData Class Reference	32
5.38	FileNode Class Reference	32
5.39	FilePlugin Class Reference	33
5.40	ARex::FileRecord Class Reference	33
5.41	FileRoot Class Reference	33
5.42	GACLPlugin Class Reference	33
5.43	gm_dirs_ Struct Reference	34
5.44	GEnvironment Class Reference	34
5.44.1	Member Function Documentation	34
5.44.1.1	nordugrid_config_loc	34
5.44.1.2	support_mail_address	34
5.45	gridftpd::GEnvironment Class Reference	35
5.45.1	Member Function Documentation	35
5.45.1.1	nordugrid_config_loc	35
5.45.1.2	support_mail_address	35
5.46	GridFTP_Commands Class Reference	35
5.47	GridFTP_Commands_timeout Class Reference	35
5.48	ARex::GridManager Class Reference	36
5.49	AuthUser::group_t Class Reference	36
5.50	Hopi::Hopi Class Reference	36
5.51	Identity Class Reference	36
5.52	IdentityGACL Class Reference	36
5.53	IdentityItemDN Class Reference	37
5.54	IdentityItemVOMS Class Reference	37
5.55	Index Class Reference	37
5.56	ISIS::ISIService Class Reference	38
5.57	ISIS::ISISecAttr Class Reference	38
5.58	Identity::Item Class Reference	38
5.59	ObjectAccess::Item Class Reference	38
5.60	ARex::FileRecord::Iterator Class Reference	39
5.61	Janitor Class Reference	39
5.61.1	Detailed Description	39
5.61.2	Constructor & Destructor Documentation	39

5.61.2.1	Janitor	39
5.61.3	Member Function Documentation	39
5.61.3.1	deploy	39
5.61.3.2	remove	40
5.61.3.3	result	40
5.61.3.4	wait	40
5.62	job_state_rec_t Struct Reference	40
5.63	JobDescription Class Reference	40
5.64	ARex::JobIDGenerator Class Reference	40
5.65	ARex::JobIDGeneratorARC Class Reference	41
5.66	ARex::JobIDGeneratorES Class Reference	41
5.67	JobLocalDescription Class Reference	41
5.68	JobLog Class Reference	42
5.68.1	Detailed Description	42
5.69	JobPlugin Class Reference	42
5.70	JobsList Class Reference	42
5.71	JobsListConfig Class Reference	42
5.71.1	Detailed Description	43
5.72	JobUser Class Reference	43
5.73	JobUserHelper Class Reference	43
5.74	JobUsers Class Reference	43
5.75	gridftp::LdapQuery Class Reference	43
5.75.1	Detailed Description	44
5.75.2	Member Enumeration Documentation	44
5.75.2.1	Scope	44
5.75.3	Constructor & Destructor Documentation	44
5.75.3.1	LdapQuery	44
5.75.3.2	~LdapQuery	44
5.75.4	Member Function Documentation	44
5.75.4.1	Host	44
5.75.4.2	Query	45
5.75.4.3	Result	45
5.76	gridftp::LdapQueryError Class Reference	45
5.76.1	Detailed Description	45

5.76.2	Constructor & Destructor Documentation	45
5.76.2.1	LdapQueryError	45
5.77	RunPlugin::lib_plugin_t Union Reference	45
5.78	gridftp::RunPlugin::lib_plugin_t Union Reference	46
5.79	LRMSResult Class Reference	46
5.80	ISIS::Neighbor_Container Class Reference	46
5.81	numvalue_for_shell Class Reference	46
5.82	ObjectAccess Class Reference	46
5.83	ObjectAccessGACL Class Reference	47
5.84	ARex::OptimizedInformationContainer Class Reference	47
5.85	gridftp::ParallelLdapQueries Class Reference	47
5.85.1	Detailed Description	47
5.86	ARex::PayloadBigFile Class Reference	47
5.86.1	Constructor & Destructor Documentation	48
5.86.1.1	PayloadBigFile	48
5.86.1.2	~PayloadBigFile	48
5.87	Hopi::PayloadBigFile Class Reference	48
5.87.1	Constructor & Destructor Documentation	48
5.87.1.1	PayloadBigFile	48
5.87.1.2	~PayloadBigFile	48
5.88	ARex::PayloadFAFile Class Reference	48
5.88.1	Constructor & Destructor Documentation	49
5.88.1.1	PayloadFAFile	49
5.89	Hopi::PayloadFile Class Reference	49
5.89.1	Detailed Description	49
5.89.2	Constructor & Destructor Documentation	49
5.89.2.1	PayloadFile	49
5.89.2.2	~PayloadFile	49
5.90	ARex::PayloadFile Class Reference	49
5.90.1	Detailed Description	50
5.90.2	Constructor & Destructor Documentation	50
5.90.2.1	PayloadFile	50
5.90.2.2	~PayloadFile	50
5.91	DREService::PerlProcessor Class Reference	50

5.91.1 Constructor & Destructor Documentation	50
5.91.1.1 PerlProcessor	50
5.91.1.2 ~PerlProcessor	51
5.92 Permission Class Reference	51
5.93 PermissionGACL Class Reference	51
5.94 Policy Class Reference	51
5.95 ArcSec::Charon::PolicyLocation Class Reference	52
5.96 ContinuationPlugins::result_t Class Reference	52
5.97 RunParallel Class Reference	52
5.98 gridftpd::RunPlugin Class Reference	52
5.99 RunPlugin Class Reference	52
5.100RunPlugins Class Reference	53
5.101RunRedirected Class Reference	53
5.102Server Class Reference	53
5.103FileRoot::ServerParams Class Reference	53
5.104ArcSec::Service_AA Class Reference	53
5.104.1 Detailed Description	53
5.105Arc::Service_JavaWrapper Class Reference	54
5.105.1 Member Function Documentation	54
5.105.1.1 process	54
5.106Arc::Service_PythonWrapper Class Reference	54
5.106.1 Member Function Documentation	54
5.106.1.1 process	54
5.107ArcSec::Service_SLCS Class Reference	54
5.107.1 Detailed Description	54
5.108SPService::Service_SP Class Reference	55
5.108.1 Detailed Description	55
5.108.2 Constructor & Destructor Documentation	55
5.108.2.1 Service_SP	55
5.108.3 Member Function Documentation	55
5.108.3.1 process	55
5.109SimpleMap Class Reference	56
5.110AuthUser::source_t Struct Reference	56
5.111UnixMap::source_t Struct Reference	56

5.112	StagingConfig Class Reference	56
5.112.1	Detailed Description	56
5.112.2	Constructor & Destructor Documentation	56
5.112.2.1	StagingConfig	56
5.113	DREService::Task Class Reference	57
5.113.1	Constructor & Destructor Documentation	57
5.113.1.1	Task	57
5.113.1.2	~Task	57
5.114	DREService::TaskQueue Class Reference	57
5.114.1	Constructor & Destructor Documentation	57
5.114.1.1	TaskQueue	57
5.114.1.2	~TaskQueue	58
5.114.2	Member Function Documentation	58
5.114.2.1	pushTask	58
5.114.2.2	shiftTask	58
5.115	DREService::TaskSet Class Reference	58
5.115.1	Constructor & Destructor Documentation	58
5.115.1.1	TaskSet	58
5.115.1.2	~TaskSet	58
5.115.2	Member Function Documentation	58
5.115.2.1	removeTask	58
5.116	DREService::PerlProcessor::ThreadInterface Struct Reference	59
5.117	UnixMap::unix_user_t Class Reference	59
5.118	UnixMap Class Reference	59
5.119	gridftp::UrlMapConfig Class Reference	59
5.120	UrlMapConfig Class Reference	59
5.121	userspec_t Class Reference	60
5.122	value_for_shell Class Reference	60
5.123	voms Struct Reference	60
5.123.1	Detailed Description	60
5.123.2	Field Documentation	60
5.123.2.1	attrs	60
5.123.2.2	server	60
5.123.2.3	voname	60

5.124voms_attrs Struct Reference	61
5.124.1 Detailed Description	61
5.124.2 Field Documentation	61
5.124.2.1 cap	61
5.124.2.2 group	61
5.124.2.3 role	61
5.125ZeroUInt Class Reference	61
5.125.1 Detailed Description	61

Chapter 1

Namespace Index

1.1 Namespace List

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

DREService	11
--------------------------------------	----

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

ARex::ARexGMConfig	13
ARex::ARexJob	13
ARex::ARexService	17
AuthEvaluator	17
AuthUser	17
AuthVO	17
CacheConfig	18
CacheConfigException	19
Cache::CacheService	19
ArcSec::Charon	21
GridFTP_Commands::close_semaphore_t	21
ContinuationPlugins::command_t	21
CommFIFO	22
gridftpd::ConfigSections	22
ConfigSections	22
ARex::DelegationStore::Consumer	22
ContinuationPlugins	22
ARex::CountedResource	23
gridftpd::Daemon	23
GridFTP_Commands::data_buffer_t	23
DataStaging::DataDeliveryService	23
ARex::DelegationStore	24
ARex::DelegationStores	24
DirectAccess::diraccess_t	24
DirectAccess	25
DirEntry	25
DREService::DREWebService	25
DTRGenerator	27
DTRInfo	30

CommFIFO::elem_t	30
Entry	30
Exec	31
ARex::FileChunks	31
ARex::FileChunksList	32
ARex::FileChunksRef	32
FileData	32
FileNode	32
FilePlugin	33
DirectFilePlugin	25
GACLPlugin	33
JobPlugin	42
ARex::FileRecord	33
FileRoot	33
gm_dirs_	34
GMEEnvironment	34
gridftp::GMEEnvironment	35
GridFTP_Commands	35
GridFTP_Commands_timeout	35
ARex::GridManager	36
AuthUser::group_t	36
Hopi::Hopi	36
Identity	36
IdentityGACL	36
Index	37
ISIS::ISISService	38
ISIS::ISISSecAttr	38
Identity::Item	38
IdentityItemDN	37
IdentityItemVOMS	37
ObjectAccess::Item	38
ARex::FileRecord::Iterator	39
Janitor	39
job_state_rec_t	40
JobDescription	40
ARex::JobIDGenerator	40
ARex::JobIDGeneratorARC	41
ARex::JobIDGeneratorES	41
JobLocalDescription	41
JobLog	42
JobsList	42
JobsListConfig	42
JobUser	43
JobUserHelper	43
JobUsers	43
gridftp::LdapQuery	43
gridftp::LdapQueryError	45
RunPlugin::lib_plugin_t	45

gridftpd::RunPlugin::lib_plugin_t	46
LRMSResult	46
ISIS::Neighbor_Container	46
numvalue_for_shell	46
ObjectAccess	46
ObjectAccessGACL	47
ARex::OptimizedInformationContainer	47
gridftpd::ParallelLdapQueries	47
ARex::PayloadBigFile	47
Hopi::PayloadBigFile	48
ARex::PayloadFAFile	48
Hopi::PayloadFile	49
ARex::PayloadFile	49
DREService::PerlProcessor	50
Permission	51
PermissionGACL	51
Policy	51
ArcSec::Charon::PolicyLocation	52
ContinuationPlugins::result_t	52
RunParallel	52
gridftpd::RunPlugin	52
RunPlugin	52
RunPlugins	53
RunRedirected	53
Server	53
FileRoot::ServerParams	53
ArcSec::Service_AA	53
Arc::Service_JavaWrapper	54
Arc::Service_PythonWrapper	54
ArcSec::Service_SLCS	54
SPService::Service_SP	55
SimpleMap	56
AuthUser::source_t	56
UnixMap::source_t	56
StagingConfig	56
DREService::Task	57
DREService::TaskQueue	57
DREService::TaskSet	58
DREService::PerlProcessor::ThreadInterface	59
UnixMap::unix_user_t	59
UnixMap	59
gridftpd::UrlMapConfig	59
UrlMapConfig	59
userspec_t	60
value_for_shell	60
voms	60
voms_attrs	61
ZeroUInt	61

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

ARex::ARexGMConfig	13
ARex::ARexJob	13
ARex::ARexService	17
AuthEvaluator	17
AuthUser	17
AuthVO	17
CacheConfig	18
CacheConfigException	19
Cache::CacheService	19
ArcSec::Charon	21
GridFTP_Commands::close_semaphore_t	21
ContinuationPlugins::command_t	21
CommFIFO	22
gridftpd::ConfigSections	22
ConfigSections	22
ARex::DelegationStore::Consumer	22
ContinuationPlugins	22
ARex::CountedResource	23
gridftpd::Daemon	23
GridFTP_Commands::data_buffer_t	23
DataStaging::DataDeliveryService	
Service for the Delivery layer of data staging	23
ARex::DelegationStore	24
ARex::DelegationStores	24
DirectAccess::diraccess_t	24
DirectAccess	25
DirectFilePlugin	25
DirEntry	25
DREService::DREWebService	25

DTRGenerator	27
DTRInfo	30
CommFIFO::elem_t	30
Entry	30
Exec	31
ARex::FileChunks	
Representation of delivered file chunks	31
ARex::FileChunksList	
Container for FileChunks instances	32
ARex::FileChunksRef	32
FileData	32
FileNode	32
FilePlugin	33
ARex::FileRecord	33
FileRoot	33
GACLPlugin	33
gm_dirs_	34
GMEnvironment	34
gridftpd::GMEnvironment	35
GridFTP_Commands	35
GridFTP_Commands_timeout	35
ARex::GridManager	36
AuthUser::group_t	36
Hopi::Hopi	36
Identity	36
IdentityGACL	36
IdentityItemDN	37
IdentityItemVOMS	37
Index	37
ISIS::ISIService	38
ISIS::ISISSecAttr	38
Identity::Item	38
ObjectAccess::Item	38
ARex::FileRecord::Iterator	39
Janitor	
Class to communicate with Janitor - Dynmaic Runtime Environment	
handler	39
job_state_rec_t	40
JobDescription	40
ARex::JobIDGenerator	40
ARex::JobIDGeneratorARC	41
ARex::JobIDGeneratorES	41
JobLocalDescription	41
JobLog	42
JobPlugin	42
JobsList	42
JobsListConfig	42
JobUser	43
JobUserHelper	43
JobUsers	43

gridftpd::LdapQuery	43
gridftpd::LdapQueryError	45
RunPlugin::lib_plugin_t	45
gridftpd::RunPlugin::lib_plugin_t	46
LRMSResult	46
ISIS::Neighbor_Container	46
numvalue_for_shell	46
ObjectAccess	46
ObjectAccessGACL	47
ARex::OptimizedInformationContainer	47
gridftpd::ParallelLdapQueries	47
ARex::PayloadBigFile	47
Hopi::PayloadBigFile	48
ARex::PayloadFAFile	48
Hopi::PayloadFile	49
ARex::PayloadFile	49
DREService::PerlProcessor	50
Permission	51
PermissionGACL	51
Policy	51
ArcSec::Charon::PolicyLocation	52
ContinuationPlugins::result_t	52
RunParallel	52
gridftpd::RunPlugin	52
RunPlugin	52
RunPlugins	53
RunRedirected	53
Server	53
FileRoot::ServerParams	53
ArcSec::Service_AA	53
Arc::Service_JavaWrapper	54
Arc::Service_PythonWrapper	54
ArcSec::Service_SLCS	54
SPService::Service_SP	55
SimpleMap	56
AuthUser::source_t	56
UnixMap::source_t	56
StagingConfig	
Represents configuration of DTR data staging	56
DREService::Task	57
DREService::TaskQueue	57
DREService::TaskSet	58
DREService::PerlProcessor::ThreadInterface	59
UnixMap::unix_user_t	59
UnixMap	59
gridftpd::UrlMapConfig	59
UrlMapConfig	59
userspec_t	60
value_for_shell	60
voms	60

voms_attrs	61
ZeroUInt	61

Chapter 4

Namespace Documentation

4.1 DREService Namespace Reference

Data Structures

- class [DREWebService](#)
- class [PerlProcessor](#)
- class [Task](#)
- class [TaskQueue](#)
- class [TaskSet](#)

4.1.1 Detailed Description

Implementation of a simple echo service

The reply of the echo service contains the string which was send to it.

Chapter 5

Data Structure Documentation

5.1 ARex::ARexGMConfig Class Reference

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

- job.h

5.2 ARex::ARexJob Class Reference

```
#include <job.h>
```

Public Member Functions

- [ARexJob](#) (const std::string &id, [ARexGMConfig](#) &config, Arc::Logger &logger, bool fast_auth_check=false)
- [ARexJob](#) (Arc::XMLNode jsdl, [ARexGMConfig](#) &config, const std::string &credentials, const std::string &clientid, Arc::Logger &logger, [JobIDGenerator](#) &idgenerator, Arc::XMLNode migration=Arc::XMLNode())
- std::string [Failure](#) (void)
- std::string [ID](#) (void)
- bool [GetDescription](#) (Arc::XMLNode &jsdl)
- bool [Cancel](#) (void)
- bool [Clean](#) (void)
- bool [Resume](#) (void)
- std::string [State](#) (void)
- std::string [State](#) (bool &job_pending)
- bool [Failed](#) (void)
- std::string [FailedState](#) (std::string &cause)
- Arc::Time [Created](#) (void)
- Arc::Time [Modified](#) (void)

- std::string [SessionDir](#) (void)
- std::string [LogDir](#) (void)
- Arc::FileAccess * [CreateFile](#) (const std::string &filename)
- Arc::FileAccess * [OpenFile](#) (const std::string &filename, bool for_read, bool for_write)
- int [OpenLogFile](#) (const std::string &name)
- Arc::FileAccess * [OpenDir](#) (const std::string &dirname)
- std::list< std::string > [LogFiles](#) (void)
- bool [UpdateCredentials](#) (const std::string &credentials)
- bool [ChooseSessionDir](#) (const std::string &jobid, std::string &sessiondir)

Static Public Member Functions

- static int [TotalJobs](#) ([ARexGMConfig](#) &config, Arc::Logger &logger)
- static std::list< std::string > [Jobs](#) ([ARexGMConfig](#) &config, Arc::Logger &logger)

5.2.1 Detailed Description

This class represents convenience interface to manage jobs handled by Grid Manager. It works mostly through corresponding classes and functions of Grid Manager.

5.2.2 Constructor & Destructor Documentation

- 5.2.2.1 [ARex::ARexJob::ARexJob](#) (const std::string & *id*, [ARexGMConfig](#) & *config*, Arc::Logger & *logger*, bool *fast_auth_check* = false)

Create instance which is an interface to existing job

- 5.2.2.2 [ARex::ARexJob::ARexJob](#) (Arc::XMLNode *jsdl*, [ARexGMConfig](#) & *config*, const std::string & *credentials*, const std::string & *clientid*, Arc::Logger & *logger*, [JobIDGenerator](#) & *idgenerator*, Arc::XMLNode *migration* = Arc::XMLNode())

Create new job with provided JSDL description

5.2.3 Member Function Documentation

- 5.2.3.1 bool [ARex::ARexJob::Cancel](#) (void)

Cancel processing/execution of job

- 5.2.3.2 bool [ARex::ARexJob::ChooseSessionDir](#) (const std::string & *jobid*, std::string & *sessiondir*)

Select a session dir to use for this job

5.2.3.3 bool ARex::ARexJob::Clean (void)

Remove job from local pool

5.2.3.4 Arc::Time ARex::ARexJob::Created (void)

Returns time when job was created.

5.2.3.5 Arc::FileAccess* ARex::ARexJob::CreateFile (const std::string & *filename*)

Creates file in job's session directory and returns handler

5.2.3.6 bool ARex::ARexJob::Failed (void)

Returns true if job has failed

5.2.3.7 std::string ARex::ARexJob::FailedState (std::string & *cause*)

Returns state at which job failed and sets cause to information what caused job failure: "internal" for server initiated and "client" for canceled on client request.

5.2.3.8 std::string ARex::ARexJob::Failure (void) [inline]

Returns textual description of failure of last operation

5.2.3.9 bool ARex::ARexJob::GetDescription (Arc::XMLNode & *jsdl*)

Fills provided jsdl with job description

5.2.3.10 std::string ARex::ARexJob::ID (void) [inline]

Return ID assigned to job

5.2.3.11 static std::list<std::string> ARex::ARexJob::Jobs (ARexGMConfig & *config*, Arc::Logger & *logger*) [static]

Returns list of user's jobs. Fine-grained ACL is ignored.

5.2.3.12 std::string ARex::ARexJob::LogDir (void)

Returns name of virtual log directory

5.2.3.13 `std::list<std::string> ARex::ARexJob::LogFiles (void)`

Returns list of existing log files

5.2.3.14 `Arc::Time ARex::ARexJob::Modified (void)`

Returns time when job state was last modified.

5.2.3.15 `Arc::FileAccess* ARex::ARexJob::OpenDir (const std::string & dirname)`

Opens directory inside session directory

5.2.3.16 `Arc::FileAccess* ARex::ARexJob::OpenFile (const std::string & filename, bool for_read, bool for_write)`

Opens file in job's session directory and returns handler

5.2.3.17 `int ARex::ARexJob::OpenLogFile (const std::string & name)`

Opens log file in control directory

5.2.3.18 `bool ARex::ARexJob::Resume (void)`

Resume execution of job after error

5.2.3.19 `std::string ARex::ARexJob::SessionDir (void)`

Returns path to session directory

5.2.3.20 `std::string ARex::ARexJob::State (void)`

Returns current state of job

5.2.3.21 `std::string ARex::ARexJob::State (bool & job_pending)`

Returns current state of job and sets *job_pending* to true if job is pending due to external limits

5.2.3.22 `static int ARex::ARexJob::TotalJobs (ARexGMConfig & config, Arc::Logger & logger) [static]`

Return number of jobs associated with this configuration. TODO: total for all user configurations.

5.2.3.23 `bool ARex::ARexJob::UpdateCredentials (const std::string & credentials)`

Updates job credentials

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

- `job.h`

5.3 ARex::ARexService Class Reference

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

- `arex.h`

5.4 AuthEvaluator Class Reference

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

- `auth.h`

5.5 AuthUser Class Reference

Data Structures

- class `group_t`
- struct `source_t`

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

- `auth.h`

5.6 AuthVO Class Reference

Friends

- class `AuthUser`

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

- `auth.h`

5.7 CacheConfig Class Reference

```
#include <conf_cache.h>
```

Public Member Functions

- [CacheConfig](#) (const [GMEEnvironment](#) &env, std::string username="")
- [CacheConfig](#) ()
- void [parseINIConf](#) (std::string username, [ConfigSections](#) *cf)
- void [setCacheDirs](#) (std::vector< std::string > cache_dirs)

5.7.1 Detailed Description

Reads conf file and provides methods to obtain cache info from it.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 `CacheConfig::CacheConfig (const GMEEnvironment & env, std::string username = "")`

Create a new [CacheConfig](#) instance. Read the config file and fill in private member variables with cache parameters. If different users are defined in the conf file, use the cache parameters for the given username.

5.7.2.2 `CacheConfig::CacheConfig () [inline]`

Empty [CacheConfig](#)

5.7.3 Member Function Documentation

5.7.3.1 `void CacheConfig::parseINIConf (std::string username, ConfigSections * cf)`

Parsers for the two different conf styles

5.7.3.2 `void CacheConfig::setCacheDirs (std::vector< std::string > cache_dirs) [inline]`

To allow for substitutions done during configuration

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

- `conf_cache.h`

5.8 CacheConfigException Class Reference

```
#include <conf_cache.h>
```

5.8.1 Detailed Description

Exception thrown by constructor caused by bad cache params in conf file

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

- `conf_cache.h`

5.9 Cache::CacheService Class Reference

```
#include <CacheService.h>
```

Public Member Functions

- [CacheService](#) (Arc::Config *cfg, Arc::PluginArgument *parg)
- virtual [~CacheService](#) (void)
- virtual Arc::MCC_Status [process](#) (Arc::Message &inmsg, Arc::Message &outmsg)
- bool [RegistrationCollector](#) (Arc::XMLNode &doc)
- [operator bool](#) ()
- bool [operator!](#) ()

Protected Member Functions

- Arc::MCC_Status [CacheCheck](#) (Arc::XMLNode in, Arc::XMLNode out, const [Job-User](#) &user)
- Arc::MCC_Status [CacheLink](#) (Arc::XMLNode in, Arc::XMLNode out, const [Job-User](#) &user, const Arc::User &mapped_user)

5.9.1 Detailed Description

[CacheService](#) provides functionality for A-REX cache operations that can be performed by remote clients. It currently consists of two operations: [CacheCheck](#) - allows querying of the cache for the presence of files. [CacheLink](#) - enables a running job to dynamically request cache files to be linked to its working (session) directory. This is especially useful in the case of pilot job workflows where job submission does not follow the usual ARC workflow. In order for input files to be available to jobs, the pilot job can call the cache service to prepare them. If requested files are not present in the cache, they can be downloaded by the cache service if requested, using the A-REX downloader utility.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Cache::CacheService::CacheService (Arc::Config * *cfg*, Arc::PluginArgument * *parg*)

Make a new [CacheService](#). Reads the configuration and determines the validity of the service.

5.9.2.2 virtual Cache::CacheService::~~CacheService (void) [virtual]

Destroy the [CacheService](#)

5.9.3 Member Function Documentation

5.9.3.1 Arc::MCC_Status Cache::CacheService::CacheCheck (Arc::XMLNode *in*, Arc::XMLNode *out*, const JobUser & *user*) [protected]

Check whether the URLs supplied in the input are present in any cache. Returns in the out message for each file true or false, and if true, the size of the file on cache disk.

Parameters

<i>user</i>	A-REX user configuration for the mapped user
-------------	--

5.9.3.2 Arc::MCC_Status Cache::CacheService::CacheLink (Arc::XMLNode *in*, Arc::XMLNode *out*, const JobUser & *user*, const Arc::User & *mapped_user*) [protected]

This method is used to link cache files to the session dir. A list of URLs is supplied and if they are present in the cache and the user calling the service has permission to access them, then they are linked to the given session directory. If the user requests that missing files be staged, then a downloader process is launched to obtain them.

Parameters

<i>user</i>	A-REX user configuration for the mapped user
<i>mapped_user</i>	The local user to which the client DN was mapped

5.9.3.3 Cache::CacheService::operator bool (void) [inline]

Returns true if the [CacheService](#) is valid.

5.9.3.4 bool Cache::CacheService::operator! (void) [inline]

Returns true if the [CacheService](#) is not valid.

5.9.3.5 `virtual Arc::MCC_Status Cache::CacheService::process (Arc::Message & inmsg,
Arc::Message & outmsg) [virtual]`

Main method called by HED when [CacheService](#) is invoked. Directs call to appropriate [CacheService](#) method.

5.9.3.6 `bool Cache::CacheService::RegistrationCollector (Arc::XMLNode & doc)`

Supplies information on the service for use in the information system.

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

- [CacheService.h](#)

5.10 ArcSec::Charon Class Reference

```
#include <charon.h>
```

Data Structures

- class [PolicyLocation](#)

5.10.1 Detailed Description

A Service which includes the ArcPDP functionality; it can be deployed as an independent service to provide request evaluation functionality for the other remote services

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

- [charon.h](#)

5.11 GridFTP_Commands::close_semaphore_t Class Reference

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

- [commands.h](#)

5.12 ContinuationPlugins::command_t Class Reference

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

- [plugins.h](#)

5.13 CommFIFO Class Reference

Data Structures

- class [elem_t](#)

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

- commfifo.h

5.14 gridftpd::ConfigSections Class Reference

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

- gridftpd/conf/conf_sections.h

5.15 ConfigSections Class Reference

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

- a-rex/grid-manager/conf/conf_sections.h

5.16 ARex::DelegationStore::Consumer Class Reference

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

- DelegationStore.h

5.17 ContinuationPlugins Class Reference

Data Structures

- class [command_t](#)
- class [result_t](#)

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

- plugins.h

5.18 ARex::CountedResource Class Reference

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

- arex.h

5.19 gridftpd::Daemon Class Reference

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

- daemon.h

5.20 GridFTP_Commands::data_buffer_t Struct Reference

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

- commands.h

5.21 DataStaging::DataDeliveryService Class Reference

```
#include <DataDeliveryService.h>
```

Public Member Functions

- [DataDeliveryService](#) (Arc::Config *cfg, Arc::PluginArgument *parg)
- virtual [~DataDeliveryService](#) ()
- virtual Arc::MCC_Status [process](#) (Arc::Message &inmsg, Arc::Message &outmsg)
- virtual void [receiveDTR](#) (DTR_ptr dtr)
- bool [RegistrationCollector](#) (Arc::XMLNode &doc)

5.21.1 Detailed Description

Service for the Delivery layer of data staging.

This service starts and controls data transfers. It assumes that the files in any request submitted are ready for immediate transfer and so do not need to be resolved or prepared in any way.

It implements DTRCallback to get callbacks when a DTR has finished transfer.

Status codes in results returned:

- OK - successful submission/cancellation

- TRANSFERRING - transfer still ongoing
- TRANSFERRED - transfer finished successfully
- TRANSFER_ERROR - transfer failed
- SERVICE_ERROR - something went wrong in the service itself

An internal list of active transfers is held in memory. After the first query of a finished transfer (successful or not) the DTR is moved to an archived list where only summary information is kept about the transfer (DTR ID, state and short error description). The DTR object is then deleted. This archived list is also kept in memory. In case a transfer is never queried, a separate thread moves any transfers which completed more than one hour ago to the archived list.

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

- DataDeliveryService.h

5.22 ARex::DelegationStore Class Reference

Data Structures

- class [Consumer](#)

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

- DelegationStore.h

5.23 ARex::DelegationStores Class Reference

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

- DelegationStores.h

5.24 DirectAccess::diraccess_t Struct Reference

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

- fileplugin.h

5.25 DirectAccess Class Reference

Data Structures

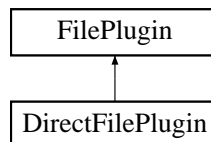
- struct [diraccess_t](#)

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

- fileplugin.h

5.26 DirectFilePlugin Class Reference

Inheritance diagram for DirectFilePlugin:



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

- fileplugin.h

5.27 DirEntry Class Reference

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

- fileroot.h

5.28 DREService::DREWebService Class Reference

Public Member Functions

- [DREWebService](#) (Arc::Config *cfg)
- virtual [~DREWebService](#) (void)
- virtual Arc::MCC_Status [process](#) (Arc::Message &inmsg, Arc::Message &outmsg)

Protected Member Functions

- Arc::MCC_Status [makeFault](#) (Arc::Message &outmsg, const std::string &reason)

Protected Attributes

- Arc::NS [ns_](#)

Static Protected Attributes

- static Arc::Logger [logger](#)

5.28.1 Constructor & Destructor Documentation

5.28.1.1 DREService::DREWebService::DREWebService (Arc::Config * *cfg*)

Constructor which is capable to extract prefix and suffix for the echo service.

5.28.1.2 virtual DREService::DREWebService::~~DREWebService (void) [virtual]

Destructor.

5.28.2 Member Function Documentation

5.28.2.1 Arc::MCC_Status DREService::DREWebService::makeFault (Arc::Message & *outmsg*, const std::string & *reason*) [protected]

Method to return an error. Creates a fault message and returns a status.

Parameters

<i>outmsg</i>	outgoing message
---------------	------------------

Returns

Status of the result achieved

5.28.2.2 virtual Arc::MCC_Status DREService::DREWebService::process (Arc::Message & *inmsg*, Arc::Message & *outmsg*) [virtual]

Implementation of the virtual method defined in MCCInterface (to be found in MCC.h).

Parameters

<i>inmsg</i>	incoming message
<i>inmsg</i>	outgoing message

Returns

Status of the result achieved

5.28.3 Field Documentation

5.28.3.1 Arc::Logger DREService::DREWebService::logger [static, protected]

Arc-intern logger. Generates output into the file specified in the arched configuration file used to invoke arched services.

5.28.3.2 Arc::NS DREService::DREWebService::ns_ [protected]

Class which specifies a XML namespace i.e. "echo". Needed to extract the content out of the incoming message

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

- dREWebService.h

5.29 DTRGenerator Class Reference

```
#include <dtr_generator.h>
```

Public Member Functions

- [DTRGenerator](#) (const [JobUsers](#) &users, void(*kicker_func)(void *)=NULL, void *kicker_arg=NULL)
- [~DTRGenerator](#) ()
- virtual void [receiveDTR](#) (DataStaging::DTR_ptr dtr)
- void [receiveJob](#) (const [JobDescription](#) &job)
- void [cancelJob](#) (const [JobDescription](#) &job)
- bool [queryJobFinished](#) ([JobDescription](#) &job)
- bool [hasJob](#) (const [JobDescription](#) &job)
- void [removeJob](#) (const [JobDescription](#) &job)
- int [checkUploadedFiles](#) ([JobDescription](#) &job)

5.29.1 Detailed Description

A-REX implementation of DTR Generator. Note that neither [Janitor](#) nor job migration functionality present in the down/uploaders has been implemented here.

5.29.2 Constructor & Destructor Documentation

5.29.2.1 **DTRGenerator::DTRGenerator** (**const JobUsers & users**, **void(*) (void *) kicker_func** = NULL, **void * kicker_arg** = NULL)

Start up Generator.

Parameters

<i>user</i>	JobUsers for this Generator.
<i>kicker_func</i>	Function to call on completion of all DTRs for a job
<i>kicker_arg</i>	Argument to kicker function

5.29.2.2 **DTRGenerator::~~DTRGenerator** ()

Stop Generator

5.29.3 Member Function Documentation

5.29.3.1 **void DTRGenerator::cancelJob** (**const JobDescription & job**)

This method is used by A-REX to cancel on-going DTRs. A cancel request is made for each DTR in the job and the method returns. The Scheduler asynchronously deals with cancelling the DTRs.

Parameters

<i>job</i>	The job which is being cancelled
------------	----------------------------------

5.29.3.2 **int DTRGenerator::checkUploadedFiles** (**JobDescription & job**)

Utility method to check that all files the user was supposed to upload with the job are ready.

Parameters

<i>job</i>	Job description, failures will be reported directly in this object.
------------	---

Returns

0 if file exists, 1 if it is not a proper file or other error, 2 if the file not there yet

5.29.3.3 **bool DTRGenerator::hasJob** (**const JobDescription & job**)

Query whether the Generator has a record of this job.

Parameters

<i>job</i>	Job to query.
------------	---------------

Returns

True if the job is active or finished.

5.29.3.4 bool DTRGenerator::queryJobFinished (JobDescription & *job*)

Query status of DTRs in job. If all DTRs are finished, returns true, otherwise returns false. If true is returned, the [JobDescription](#) should be checked for whether the staging was successful or not by checking `GetFailure()`.

Parameters

<i>job</i>	Description of job to query. Can be modified to add a failure reason.
------------	---

Returns

True if all DTRs in the job are finished, false otherwise.

5.29.3.5 virtual void DTRGenerator::receiveDTR (DataStaging::DTR_ptr *dtr*) [virtual]

Callback called when DTR is finished. This DTR is marked done in the DTR list and if all DTRs for the job have completed, the job is marked as done.

Parameters

<i>dtr</i>	DTR object sent back from the Scheduler
------------	---

5.29.3.6 void DTRGenerator::receiveJob (const JobDescription & *job*)

A-REX sends data transfer requests to the data staging system through this method. It reads the job.id.input/output files, forms DTRs and sends them to the Scheduler.

Parameters

<i>job</i>	Job description object.
------------	-------------------------

5.29.3.7 void DTRGenerator::removeJob (const JobDescription & *job*)

Remove the job from the Generator. Only finished jobs will be removed, and a warning will be logged if the job still has active DTRs. This method should be called after A-REX has finished PREPARING or FINISHING.

Parameters

<i>job</i>	The job to remove.
------------	--------------------

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

- `dtr_generator.h`

5.30 DTRInfo Class Reference

```
#include <dtr_generator.h>
```

Public Member Functions

- [DTRInfo](#) (const [JobUsers](#) &users)

5.30.1 Detailed Description

[DTRInfo](#) passes state information from data staging to A-REX via the defined callback, called when the DTR passes to the certain processes. It could for example write to files in the control directory, and this information can be picked up and published by the info system.

5.30.2 Constructor & Destructor Documentation

5.30.2.1 DTRInfo::DTRInfo (const [JobUsers](#) & *users*)

[JobUsers](#) is needed to find the correct control dir

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

- `dtr_generator.h`

5.31 CommFIFO::elem_t Class Reference

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

- `commfifo.h`

5.32 Entry Class Reference

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

- `Entry.h`

5.33 Exec Class Reference

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

- `info_types.h`

5.34 ARex::FileChunks Class Reference

```
#include <FileChunks.h>
```

Public Member Functions

- `std::string Path` (void)
- `void Size` (off_t size)
- `off_t Size` (void)
- `void Add` (off_t start, off_t csize)
- `bool Complete` (void)
- `void Print` (void)
- `void Release` (void)
- `void Remove` (void)

5.34.1 Detailed Description

Representation of delivered file chunks.

5.34.2 Member Function Documentation

5.34.2.1 void ARex::FileChunks::Release (void)

Release reference obtained through `FileChunksList::Get()` method. This operation may lead to destruction of FileChunk instance hence previously obtained reference must not be used.

5.34.2.2 void ARex::FileChunks::Remove (void)

Relases reference obtained through `Get()` method and destroys its instance. Normally this method to be called instead of `Release()` after whole file is delivered in order to free resources associated with `FileChunks` instance.

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

- `FileChunks.h`

5.35 ARex::FileChunksList Class Reference

```
#include <FileChunks.h>
```

Public Member Functions

- [FileChunks](#) & [Get](#) (std::string path)
- void [Timeout](#) (int t)

5.35.1 Detailed Description

Container for [FileChunks](#) instances.

5.35.2 Member Function Documentation

5.35.2.1 FileChunks& ARex::FileChunksList::Get (std::string *path*)

Returns previously created [FileChunks](#) object with associated path. If such instance does not exist new one is created. Obtained reference may be used for other operations. Obtained reference must be Release()ed after it is not longer needed.

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

- FileChunks.h

5.36 ARex::FileChunksRef Class Reference

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

- FileChunks.h

5.37 FileData Class Reference

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

- info_types.h

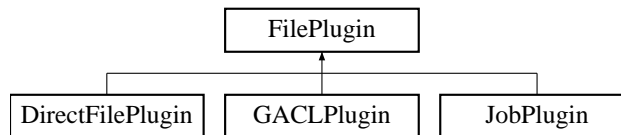
5.38 FileNode Class Reference

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

- fileroot.h

5.39 FilePlugin Class Reference

Inheritance diagram for FilePlugin:



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

- fileroot.h

5.40 ARex::FileRecord Class Reference

Data Structures

- class [Iterator](#)

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

- FileRecord.h

5.41 FileRoot Class Reference

Data Structures

- class [ServerParams](#)

Friends

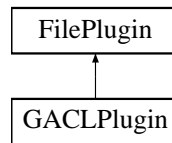
- class **GridFTP_Commands**

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

- fileroot.h

5.42 GACLPlugin Class Reference

Inheritance diagram for GACLPlugin:



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

- gacplugin.h

5.43 gm_dirs_ Struct Reference

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

- jobplugin.h

5.44 GMEEnvironment Class Reference

Public Member Functions

- std::string [nordugrid_loc](#) (void) const
- std::string [nordugrid_data_loc](#) (void) const
- std::string [nordugrid_libexec_loc](#) (void) const
- std::string [nordugrid_config_loc](#) (void) const
- std::string [support_mail_address](#) (void) const

5.44.1 Member Function Documentation

5.44.1.1 std::string GMEEnvironment::nordugrid_config_loc (void) const

ARC configuration file /etc/arc.conf \$ARC_LOCATION/etc/arc.conf

5.44.1.2 std::string GMEEnvironment::support_mail_address (void) const

Email address of person responsible for this ARC installation grid.manager, it can also be set from configuration file

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

- a-rex/grid-manager/conf/environment.h

5.45 gridftpd::GMEEnvironment Class Reference

Public Member Functions

- `std::string nordugrid_loc` (void) const
- `std::string nordugrid_libexec_loc` (void) const
- `std::string nordugrid_config_loc` (void) const
- `std::string support_mail_address` (void) const

5.45.1 Member Function Documentation

5.45.1.1 `std::string gridftpd::GMEEnvironment::nordugrid_config_loc (void) const`

ARC configuration file /etc/arc.conf \$ARC_LOCATION/etc/arc.conf

5.45.1.2 `std::string gridftpd::GMEEnvironment::support_mail_address (void) const`

Email address of person responsible for this ARC installation grid.manager, it can also be set from configuration file

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

- `gridftpd/conf/environment.h`

5.46 GridFTP_Commands Class Reference

Data Structures

- class `close_semaphore_t`
- struct `data_buffer_t`

Friends

- class `GridFTP_Commands_timeout`

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

- `commands.h`

5.47 GridFTP_Commands_timeout Class Reference

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

- `commands.h`

5.48 ARex::GridManager Class Reference

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

- grid_manager.h

5.49 AuthUser::group_t Class Reference

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

- auth.h

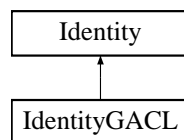
5.50 Hopi::Hopi Class Reference

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

- hopi.h

5.51 Identity Class Reference

Inheritance diagram for Identity:



Data Structures

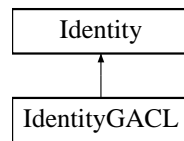
- class [Item](#)

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

- identity.h

5.52 IdentityGACL Class Reference

Inheritance diagram for IdentityGACL:

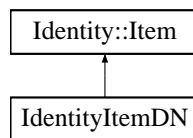


The documentation for this class was generated from the following file:

- identity_gacl.h

5.53 IdentityItemDN Class Reference

Inheritance diagram for IdentityItemDN:

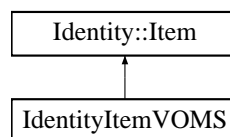


The documentation for this class was generated from the following file:

- identity_dn.h

5.54 IdentityItemVOMS Class Reference

Inheritance diagram for IdentityItemVOMS:



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

- identity_voms.h

5.55 Index Class Reference

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

- Index.h

5.56 ISIS::ISIService Class Reference

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

- isis.h

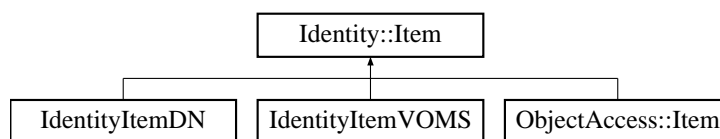
5.57 ISIS::ISISecAttr Class Reference

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

- security.h

5.58 Identity::Item Class Reference

Inheritance diagram for Identity::Item:

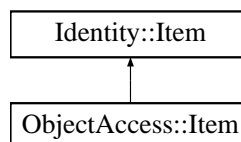


The documentation for this class was generated from the following file:

- identity.h

5.59 ObjectAccess::Item Class Reference

Inheritance diagram for ObjectAccess::Item:



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

- object_access.h

5.60 ARex::FileRecord::Iterator Class Reference

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

- FileRecord.h

5.61 Janitor Class Reference

```
#include <janitor.h>
```

Public Member Functions

- [Janitor](#) (const std::string &id, const std::string &cdir, const [GMEEnvironment](#) &env)
- bool [enabled](#) ()
- [operator bool](#) (void)
- bool [operator!](#) (void)
- bool [deploy](#) (void)
- bool [remove](#) (void)
- bool [wait](#) (int timeout)
- Result [result](#) (void)

5.61.1 Detailed Description

Class to communicate with [Janitor](#) - Dynmaic Runtime Environment handler.

5.61.2 Constructor & Destructor Documentation

5.61.2.1 [Janitor::Janitor](#) (const std::string & *id*, const std::string & *cdir*, const [GMEEnvironment](#) & *env*)

Creates instance representing job entry in [Janitor](#) database.

Takes id for job identifier and cdir for the control directory of A-Rex. constructor does not register job in the [Janitor](#). It only associates job with this instance.

5.61.3 Member Function Documentation

5.61.3.1 [bool Janitor::deploy](#) (void)

Registers associated job with [Janitor](#) and deploys dynamic RTEs.

This operation is asynchronous. Returned true means [Janitor](#) will be contacted and deployment will start soon. For obtaining result of operation see methods [wait\(\)](#) and [result\(\)](#). During this operation janitor utility is called with command register and optionally deploy.

5.61.3.2 `bool Janitor::remove (void)`

Removes job from those handled by [Janitor](#) and releases associated RTEs.

This operation is asynchronous. Returned true means [Janitor](#) will be contacted and removal will start soon. For obtaining result of operation see methods [wait\(\)](#) and [result\(\)](#). During this operation janitor utility is called with command remove.

5.61.3.3 `Result Janitor::result (void)`

Returns true if operation initiated by [deploy\(\)](#) or [remove\(\)](#) succeeded.

It should be called after [wait\(\)](#) returned true.

5.61.3.4 `bool Janitor::wait (int timeout)`

Wait till operation initiated by [deploy\(\)](#) or [remove\(\)](#) finished.

This operation returns true if operation finished or false if timeout seconds passed. It may be called repeatedly and even after it previously returned true. If no operation is running it returns true immediately.

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

- janitor.h

5.62 `job_state_rec_t` Struct Reference

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

- grid-manager/jobs/job.h

5.63 JobDescription Class Reference

Friends

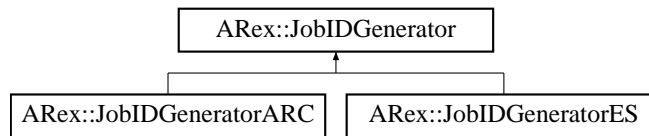
- class **JobsList**

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

- grid-manager/jobs/job.h

5.64 `ARex::JobIDGenerator` Class Reference

Inheritance diagram for `ARex::JobIDGenerator`:

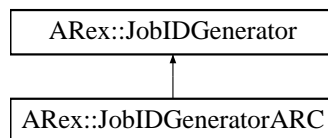


The documentation for this class was generated from the following file:

- tools.h

5.65 ARex::JobIDGeneratorARC Class Reference

Inheritance diagram for ARex::JobIDGeneratorARC:

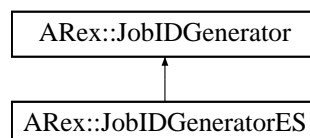


The documentation for this class was generated from the following file:

- tools.h

5.66 ARex::JobIDGeneratorES Class Reference

Inheritance diagram for ARex::JobIDGeneratorES:



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

- tools.h

5.67 JobLocalDescription Class Reference

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

- info_types.h

5.68 JobLog Class Reference

```
#include <job_log.h>
```

5.68.1 Detailed Description

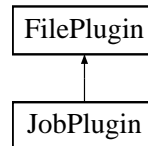
Put short information into log when every job starts/finishes. And store more detailed information for Reporter.

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

- job_log.h

5.69 JobPlugin Class Reference

Inheritance diagram for JobPlugin:



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

- jobplugin.h

5.70 JobsList Class Reference

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

- states.h

5.71 JobsListConfig Class Reference

```
#include <job_config.h>
```

Friends

- class **JobsList**

5.71.1 Detailed Description

Class to represent information read from configuration.

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

- job_config.h

5.72 JobUser Class Reference

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

- users.h

5.73 JobUserHelper Class Reference

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

- users.h

5.74 JobUsers Class Reference

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

- users.h

5.75 gridftp::LdapQuery Class Reference

```
#include <ldapquery.h>
```

Public Types

- enum [Scope](#)

Public Member Functions

- [LdapQuery](#) (const std::string &ldaphost, int ldapport, bool anonymous=true, const std::string &usersn="", int timeout=20)
- [~LdapQuery](#) ()
- void [Query](#) (const std::string &base, const std::string &filter="(objectclass=*)", const std::vector< std::string > &attributes=std::vector< std::string >(), [Scope](#) scope=subtree) throw (LdapQueryError)
- void [Result](#) (ldap_callback callback, void *ref) throw (LdapQueryError)
- std::string [Host](#) ()

5.75.1 Detailed Description

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

5.75.2 Member Enumeration Documentation

5.75.2.1 enum gridftp::LdapQuery::Scope

Scope for a LDAP queries. Use when querying.

5.75.3 Constructor & Destructor Documentation

5.75.3.1 gridftp::LdapQuery::LdapQuery (const std::string &ldaphost, int ldapport, bool anonymous = true, const std::string &usersn = " ", int timeout = 20)

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

5.75.3.2 gridftp::LdapQuery::~~LdapQuery ()

Destructor. Will disconnect from the ldapserver if still connected.

5.75.4 Member Function Documentation

5.75.4.1 std::string gridftp::LdapQuery::Host ()

Returns the hostname of the ldap-server.

5.75.4.2 void gridftpdp::LdapQuery::Query (const std::string & *base*, const std::string & *filter* = "(objectclass=*)", const std::vector< std::string > & *attributes* = std::vector< std::string >(), Scope *scope* = subtree) throw (LdapQueryError)

Queries the ldap server.

5.75.4.3 void gridftpdp::LdapQuery::Result (ldap_callback *callback*, void * *ref*) throw (LdapQueryError)

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

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

- ldapquery.h

5.76 gridftpdp::LdapQueryError Class Reference

```
#include <ldapquery.h>
```

Public Member Functions

- [LdapQueryError](#) (std::string message)

5.76.1 Detailed Description

[LdapQuery](#) exception. Gets thrown when an error occurs in a query.

5.76.2 Constructor & Destructor Documentation

5.76.2.1 gridftpdp::LdapQueryError::LdapQueryError (std::string *message*) [inline]

Standard exception class constructor.

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

- ldapquery.h

5.77 RunPlugin::lib_plugin_t Union Reference

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

- a-rex/grid-manager/run/run_plugin.h

5.78 gridftpd::RunPlugin::lib_plugin_t Union Reference

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

- gridftpd/run/run_plugin.h

5.79 LRMSResult Class Reference

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

- info_types.h

5.80 ISIS::Neighbor_Container Class Reference

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

- isis.h

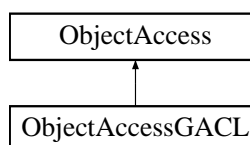
5.81 numvalue_for_shell Class Reference

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

- job_desc.h

5.82 ObjectAccess Class Reference

Inheritance diagram for ObjectAccess:



Data Structures

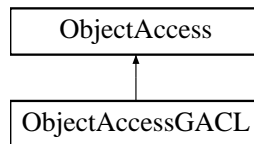
- class [Item](#)

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

- object_access.h

5.83 ObjectAccessGACL Class Reference

Inheritance diagram for ObjectAccessGACL:



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

- object_access_gacl.h

5.84 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

5.85 gridftp::ParallelLdapQueries Class Reference

```
#include <ldapquery.h>
```

5.85.1 Detailed Description

General method to perform parallel ldap-queries to a set of clusters

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

- ldapquery.h

5.86 ARex::PayloadBigFile Class Reference

Public Member Functions

- [PayloadBigFile](#) (const char *filename, Size_t start, Size_t end)
- virtual [~PayloadBigFile](#) (void)

5.86.1 Constructor & Destructor Documentation

5.86.1.1 ARex::PayloadBigFile::PayloadBigFile (const char * *filename*, Size_t *start*, Size_t *end*)

Creates object associated with file for reading from it

5.86.1.2 virtual ARex::PayloadBigFile::~~PayloadBigFile (void) [virtual]

Creates object associated with file for writing into it. Use size=-1 for undefined size.

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

- a-rex/PayloadFile.h

5.87 Hopi::PayloadBigFile Class Reference

Public Member Functions

- [PayloadBigFile](#) (const char *filename, Size_t start, Size_t end)
- virtual [~PayloadBigFile](#) (void)

5.87.1 Constructor & Destructor Documentation

5.87.1.1 Hopi::PayloadBigFile::PayloadBigFile (const char * *filename*, Size_t *start*, Size_t *end*)

Creates object associated with file for reading from it

5.87.1.2 virtual Hopi::PayloadBigFile::~~PayloadBigFile (void) [virtual]

Creates object associated with file for writing into it. Use size=-1 for undefined size.

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

- hopi/PayloadFile.h

5.88 ARex::PayloadFAFile Class Reference

Public Member Functions

- [PayloadFAFile](#) (Arc::FileAccess *h, Size_t start, Size_t end)

5.88.1 Constructor & Destructor Documentation

5.88.1.1 ARex::PayloadFAFile::PayloadFAFile (Arc::FileAccess * *h*, Size_t *start*, Size_t *end*)

Creates object associated with file for reading from it

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

- a-rex/PayloadFile.h

5.89 Hopi::PayloadFile Class Reference

```
#include <PayloadFile.h>
```

Public Member Functions

- [PayloadFile](#) (const char *filename, Size_t start, Size_t end)
- virtual [~PayloadFile](#) (void)

5.89.1 Detailed Description

Implementation of PayloadRawInterface which provides access to ordinary file. -
Currently only read-only mode is supported.

5.89.2 Constructor & Destructor Documentation

5.89.2.1 Hopi::PayloadFile::PayloadFile (const char * *filename*, Size_t *start*, Size_t *end*)

Creates object associated with file for reading from it. Use end=-1 for full size.

5.89.2.2 virtual Hopi::PayloadFile::~~PayloadFile (void) [virtual]

Creates object associated with file for writing into it. Use size=-1 for undefined size.

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

- hopi/PayloadFile.h

5.90 ARex::PayloadFile Class Reference

```
#include <PayloadFile.h>
```

Public Member Functions

- [PayloadFile](#) (const char *filename, Size_t start, Size_t end)
- virtual [~PayloadFile](#) (void)

5.90.1 Detailed Description

Implementation of PayloadRawInterface which provides access to ordinary file. -
Currently only read-only mode is supported.

5.90.2 Constructor & Destructor Documentation

5.90.2.1 ARex::PayloadFile::PayloadFile (const char * filename, Size_t start, Size_t end)

Creates object associated with file for reading from it. Use end=-1 for full size.

5.90.2.2 virtual ARex::PayloadFile::~~PayloadFile (void) [virtual]

Creates object associated with file for writing into it. Use size=-1 for undefined size.

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

- a-rex/PayloadFile.h

5.91 DREService::PerlProcessor Class Reference

Data Structures

- struct [ThreadInterface](#)

Public Member Functions

- [PerlProcessor](#) (int threadNumber, [TaskQueue](#) *pTaskQueue, [TaskSet](#) *pTaskSet)
- virtual [~PerlProcessor](#) (void)

5.91.1 Constructor & Destructor Documentation

5.91.1.1 DREService::PerlProcessor::PerlProcessor (int threadNumber, TaskQueue * pTaskQueue, TaskSet * pTaskSet)

Constructor which is capable to extract prefix and suffix for the echo service.

5.91.1.2 virtual DREService::PerlProcessor::~~PerlProcessor (void) [virtual]

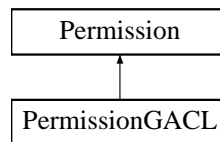
Destructor.

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

- PerlProcessor.h

5.92 Permission Class Reference

Inheritance diagram for Permission:

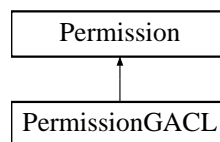


The documentation for this class was generated from the following file:

- permission.h

5.93 PermissionGACL Class Reference

Inheritance diagram for PermissionGACL:



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

- permission_gacl.h

5.94 Policy Class Reference

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

- Policy.h

5.95 ArcSec::Charon::PolicyLocation Class Reference

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

- charon.h

5.96 ContinuationPlugins::result_t Class Reference

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

- plugins.h

5.97 RunParallel Class Reference

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

- run_parallel.h

5.98 gridftpd::RunPlugin Class Reference

Data Structures

- union [lib_plugin_t](#)

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

- gridftpd/run/run_plugin.h

5.99 RunPlugin Class Reference

Data Structures

- union [lib_plugin_t](#)

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

- a-rex/grid-manager/run/run_plugin.h

5.100 RunPlugins Class Reference

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

- a-rex/grid-manager/run/run_plugin.h

5.101 RunRedirected Class Reference

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

- run_redirected.h

5.102 Server Class Reference

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

- Server.h

5.103 FileRoot::ServerParams Class Reference

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

- fileroot.h

5.104 ArcSec::Service_AA Class Reference

```
#include <aaservice.h>
```

5.104.1 Detailed Description

A Service which includes the AttributeAuthority functionality; it accepts the <samlp:-AttributeQuery> which includes the <Subject> of the principal from the request and <Attribute> which the request would get; it access some local attribute database and returns <samlp:Assertion> which includes the <Attribute>

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

- aaservice.h

5.105 Arc::Service_JavaWrapper Class Reference

Public Member Functions

- virtual Arc::MCC_Status [process](#) (Arc::Message &, Arc::Message &)

5.105.1 Member Function Documentation

- 5.105.1.1 virtual Arc::MCC_Status Arc::Service_JavaWrapper::process (Arc::Message & , Arc::Message &) [virtual]

Service request processing routine

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

- javawrapper.h

5.106 Arc::Service_PythonWrapper Class Reference

Public Member Functions

- virtual Arc::MCC_Status [process](#) (Arc::Message &, Arc::Message &)

5.106.1 Member Function Documentation

- 5.106.1.1 virtual Arc::MCC_Status Arc::Service_PythonWrapper::process (Arc::Message & , Arc::Message &) [virtual]

Service request processing routine

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

- pythonwrapper.h

5.107 ArcSec::Service_SLCS Class Reference

```
#include <slcs.h>
```

5.107.1 Detailed Description

A Service which signs the short-lived certificate; it accepts the certificate signing request (CSR) from client side through soap, signs a short-lived certificate and sends back through soap. This service is supposed to be deployed together with the SPService

and saml2sso.serviceprovider handler, in order to sign certificate based on the authentication result from saml2sso profile. Also the saml attribute (inside the saml assertion from saml2sso profile) will be put into the signed short-lived certificate. By deploying this service together with SPService and saml2sso.serviceprovider handler, we can get the conversion from username/password -----> x509 certificate.

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

- slcs.h

5.108 SPService::Service_SP Class Reference

```
#include <SPService.h>
```

Public Member Functions

- [Service_SP](#) (Arc::Config *cfg)
- virtual Arc::MCC_Status [process](#) (Arc::Message &, Arc::Message &)

5.108.1 Detailed Description

This is service which accepts HTTP request from user agent (web browser) in the client side and processes the functionality of Service Provider in SAML2 SSO profile --- composing <AuthnRequest> Note: the IdP name is provided by the user agent directly when it gives a request, instead of the WRYF(where are you from) or Discovery Service in other implementation

5.108.2 Constructor & Destructor Documentation

5.108.2.1 SPService::Service_SP::Service_SP (Arc::Config * cfg)

Constructor

5.108.3 Member Function Documentation

5.108.3.1 virtual Arc::MCC_Status SPService::Service_SP::process (Arc::Message & , Arc::Message &) [virtual]

Service request processing routine

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

- SPService.h

5.109 SimpleMap Class Reference

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

- `simplemap.h`

5.110 AuthUser::source_t Struct Reference

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

- `auth.h`

5.111 UnixMap::source_t Struct Reference

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

- `unixmap.h`

5.112 StagingConfig Class Reference

```
#include <conf_staging.h>
```

Public Member Functions

- [StagingConfig](#) (const [GMEEnvironment](#) &env)

Friends

- class **DTRGenerator**

5.112.1 Detailed Description

Represents configuration of DTR data staging.

5.112.2 Constructor & Destructor Documentation

5.112.2.1 StagingConfig::StagingConfig (const GMEEnvironment & env)

Load config from configuration file. Information from [JobsListConfig](#) is used first, then it is overwritten by parameters in [data-staging] (for ini style) or new staging parameters in <dataTransfer> (for xml style).

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

- `conf_staging.h`

5.113 DREService::Task Class Reference

Public Member Functions

- [Task](#) (int taskID, Arc::Message *request, Arc::Message *response)
- virtual [~Task](#) (void)

5.113.1 Constructor & Destructor Documentation

5.113.1.1 `DREService::Task::Task (int taskID, Arc::Message * request, Arc::Message * response)`

Constructor which is capable to extract prefix and suffix for the echo service.

5.113.1.2 `virtual DREService::Task::~Task (void)` [virtual]

Destructor.

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

- `Task.h`

5.114 DREService::TaskQueue Class Reference

Public Member Functions

- [TaskQueue](#) (int length)
- virtual [~TaskQueue](#) (void)
- int [pushTask](#) ([Task](#) *task)
- [Task](#) * [shiftTask](#) ()

5.114.1 Constructor & Destructor Documentation

5.114.1.1 `DREService::TaskQueue::TaskQueue (int length)`

Constructor which is capable to extract prefix and suffix for the echo service.

5.114.1.2 `virtual DREService::TaskQueue::~~TaskQueue (void) [virtual]`

Destructor.

5.114.2 Member Function Documentation

5.114.2.1 `int DREService::TaskQueue::pushTask (Task * task)`

Blocks, if taskqueue is full. If task is stored in the queue and had a taskID == -1 it gets a fresh taskID.

5.114.2.2 `Task* DREService::TaskQueue::shiftTask ()`

Shifts the first task from the queue (and removes it).

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

- TaskQueue.h

5.115 DREService::TaskSet Class Reference

Public Member Functions

- [TaskSet](#) (int size)
- `virtual ~TaskSet` (void)
- `Task * removeTask` (int)

5.115.1 Constructor & Destructor Documentation

5.115.1.1 `DREService::TaskSet::TaskSet (int size)`

Constructor which is capable to extract prefix and suffix for the echo service.

5.115.1.2 `virtual DREService::TaskSet::~~TaskSet (void) [virtual]`

Destructor.

5.115.2 Member Function Documentation

5.115.2.1 `Task* DREService::TaskSet::removeTask (int)`

Checks wheter there is a task in the queue having that taskID in order to return it. If such a taskID is not available, the method blocks until such a taskID is available. The task will be removed from the stack in that case.

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

- TaskSet.h

5.116 DREService::PerlProcessor::ThreadInterface Struct Reference

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

- PerlProcessor.h

5.117 UnixMap::unix_user_t Class Reference

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

- unixmap.h

5.118 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

5.119 gridftpd::UrlMapConfig Class Reference

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

- gridftpd/conf/conf_map.h

5.120 UrlMapConfig Class Reference

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

- a-rex/grid-manager/conf/conf_map.h

5.121 userspec_t Class Reference

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

- userspec.h

5.122 value_for_shell Class Reference

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

- job_desc.h

5.123 voms Struct Reference

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

Data Fields

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

5.123.1 Detailed Description

VOMS data

5.123.2 Field Documentation

5.123.2.1 std::vector<voms_attrs> voms::attrs

User's characteristics

5.123.2.2 std::string voms::server

The VOMS server DN, as from its certificate

5.123.2.3 std::string voms::vname

The name of the VO to which the VOMS belongs

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

- auth.h

5.124 voms_attrs Struct Reference

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

Data Fields

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

5.124.1 Detailed Description

VOMS attributes

5.124.2 Field Documentation

5.124.2.1 `std::string voms_attrs::cap`

user's capability

5.124.2.2 `std::string voms_attrs::group`

user's group

5.124.2.3 `std::string voms_attrs::role`

user's role

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

- `auth.h`

5.125 ZeroUInt Class Reference

```
#include <job_config.h>
```

5.125.1 Detailed Description

[ZeroUInt](#) is a wrapper around unsigned int. It provides a consistent default value, as int type variables have no predefined value assigned upon creation. It also protects from potential counter underflow, to stop counter jumping to MAX_INT.

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

- `job_config.h`