

Hosting Environment (Daemon) Services

Generated by Doxygen 1.7.5

Thu Mar 1 2012 19:48:41

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	CreateFile	15
5.2.3.5	Failed	15
5.2.3.6	Failure	15

5.2.3.7	GetDescription	15
5.2.3.8	ID	15
5.2.3.9	Jobs	15
5.2.3.10	LogDir	15
5.2.3.11	LogFiles	15
5.2.3.12	OpenDir	15
5.2.3.13	OpenFile	16
5.2.3.14	OpenLogFile	16
5.2.3.15	Resume	16
5.2.3.16	SessionDir	16
5.2.3.17	State	16
5.2.3.18	State	16
5.2.3.19	TotalJobs	16
5.2.3.20	UpdateCredentials	16
5.3	ARex::ARexService Class Reference	16
5.4	AuthEvaluator Class Reference	17
5.5	AuthUser Class Reference	17
5.6	AuthVO Class Reference	17
5.7	CacheConfig Class Reference	17
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	18
5.8.1	Detailed Description	18
5.9	Cache::CacheService Class Reference	19
5.9.1	Detailed Description	19
5.9.2	Constructor & Destructor Documentation	19
5.9.2.1	CacheService	19
5.9.2.2	~CacheService	19
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	20
5.9.3.6	RegistrationCollector	20
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	21
5.14	ConfigSections Class Reference	22
5.15	gridftpd::ConfigSections Class Reference	22
5.16	ContinuationPlugins Class Reference	22
5.17	ARex::CountedResource Class Reference	22
5.18	gridftpd::Daemon Class Reference	22
5.19	GridFTP_Commands::data_buffer_t Struct Reference	22
5.20	DataStaging::DataDeliveryService Class Reference	23
5.20.1	Detailed Description	23
5.21	DirectAccess::diraccess_t Struct Reference	24
5.22	DirectAccess Class Reference	24
5.23	DirectFilePlugin Class Reference	24
5.24	DirEntry Class Reference	24
5.25	DREService::DREWebService Class Reference	24
5.25.1	Constructor & Destructor Documentation	25
5.25.1.1	DREWebService	25
5.25.1.2	~DREWebService	25
5.25.2	Member Function Documentation	25
5.25.2.1	makeFault	25
5.25.2.2	process	26
5.25.3	Field Documentation	26
5.25.3.1	logger	26
5.25.3.2	ns_	26
5.26	DTRGenerator Class Reference	26

5.26.1 Detailed Description	27
5.26.2 Constructor & Destructor Documentation	27
5.26.2.1 DTRGenerator	27
5.26.2.2 ~DTRGenerator	27
5.26.3 Member Function Documentation	27
5.26.3.1 cancelJob	27
5.26.3.2 checkUploadedFiles	28
5.26.3.3 hasJob	28
5.26.3.4 queryJobFinished	28
5.26.3.5 receiveDTR	28
5.26.3.6 receiveJob	29
5.26.3.7 removeJob	29
5.27 DTRInfo Class Reference	29
5.27.1 Detailed Description	29
5.27.2 Constructor & Destructor Documentation	29
5.27.2.1 DTRInfo	30
5.28 CommFIFO::elem_t Class Reference	30
5.29 Entry Class Reference	30
5.30 ARex::FileChunks Class Reference	30
5.30.1 Detailed Description	30
5.30.2 Member Function Documentation	31
5.30.2.1 Release	31
5.30.2.2 Remove	31
5.31 ARex::FileChunksList Class Reference	31
5.31.1 Detailed Description	31
5.31.2 Member Function Documentation	31
5.31.2.1 Get	31
5.32 ARex::FileChunksRef Class Reference	32
5.33 FileData Class Reference	32
5.34 FileNode Class Reference	32
5.35 FilePlugin Class Reference	32
5.36 FileRoot Class Reference	32
5.37 GACLPlugin Class Reference	33
5.38 gm_dirs_ Struct Reference	33

5.39	gridftpd::GMEEnvironment Class Reference	33
5.39.1	Member Function Documentation	33
5.39.1.1	nordugrid_config_loc	33
5.39.1.2	support_mail_address	34
5.40	GMEEnvironment Class Reference	34
5.40.1	Member Function Documentation	34
5.40.1.1	nordugrid_config_loc	34
5.40.1.2	support_mail_address	34
5.41	GridFTP_Commands Class Reference	34
5.42	GridFTP_Commands_timeout Class Reference	35
5.43	ARex::GridManager Class Reference	35
5.44	AuthUser::group_t Class Reference	35
5.45	Hopi::Hopi Class Reference	35
5.46	Identity Class Reference	35
5.47	IdentityGACL Class Reference	36
5.48	IdentityItemDN Class Reference	36
5.49	IdentityItemVOMS Class Reference	36
5.50	Index Class Reference	37
5.51	ISIS::ISIService Class Reference	37
5.52	ISIS::ISISecAttr Class Reference	37
5.53	ObjectAccess::Item Class Reference	37
5.54	Identity::Item Class Reference	38
5.55	Janitor Class Reference	38
5.55.1	Detailed Description	38
5.55.2	Constructor & Destructor Documentation	38
5.55.2.1	Janitor	38
5.55.3	Member Function Documentation	39
5.55.3.1	deploy	39
5.55.3.2	remove	39
5.55.3.3	result	39
5.55.3.4	wait	39
5.56	job_state_rec_t Struct Reference	39
5.57	JobDescription Class Reference	39
5.58	JobLocalDescription Class Reference	40

5.59 JobLog Class Reference	40
5.59.1 Detailed Description	40
5.60 JobPlugin Class Reference	40
5.61 ARex::JobRecord Class Reference	41
5.62 JobsList Class Reference	41
5.63 JobsListConfig Class Reference	41
5.63.1 Detailed Description	41
5.64 JobUser Class Reference	41
5.65 JobUserHelper Class Reference	41
5.66 JobUsers Class Reference	42
5.67 gridftpdp::LdapQuery Class Reference	42
5.67.1 Detailed Description	42
5.67.2 Member Enumeration Documentation	42
5.67.2.1 Scope	42
5.67.3 Constructor & Destructor Documentation	42
5.67.3.1 LdapQuery	42
5.67.3.2 ~LdapQuery	43
5.67.4 Member Function Documentation	43
5.67.4.1 Host	43
5.67.4.2 Query	43
5.67.4.3 Result	43
5.68 gridftpdp::LdapQueryError Class Reference	43
5.68.1 Detailed Description	43
5.68.2 Constructor & Destructor Documentation	43
5.68.2.1 LdapQueryError	43
5.69 RunPlugin::lib_plugin_t Union Reference	44
5.70 gridftpdp::RunPlugin::lib_plugin_t Union Reference	44
5.71 ARex::LoggerClient Class Reference	44
5.72 LRMSResult Class Reference	44
5.73 ISIS::Neighbor_Container Class Reference	44
5.74 numvalue_for_shell Class Reference	44
5.75 ObjectAccess Class Reference	45
5.76 ObjectAccessGACL Class Reference	45
5.77 ARex::OptimizedInformationContainer Class Reference	45

5.78	gridftp::ParallelLdapQueries Class Reference	45
5.78.1	Detailed Description	46
5.79	ARex::PayloadBigFile Class Reference	46
5.79.1	Constructor & Destructor Documentation	46
5.79.1.1	PayloadBigFile	46
5.79.1.2	~PayloadBigFile	46
5.80	Hopi::PayloadBigFile Class Reference	46
5.80.1	Constructor & Destructor Documentation	46
5.80.1.1	PayloadBigFile	46
5.80.1.2	~PayloadBigFile	47
5.81	ARex::PayloadFAFile Class Reference	47
5.81.1	Constructor & Destructor Documentation	47
5.81.1.1	PayloadFAFile	47
5.82	Hopi::PayloadFile Class Reference	47
5.82.1	Detailed Description	47
5.82.2	Constructor & Destructor Documentation	47
5.82.2.1	PayloadFile	47
5.82.2.2	~PayloadFile	48
5.83	ARex::PayloadFile Class Reference	48
5.83.1	Detailed Description	48
5.83.2	Constructor & Destructor Documentation	48
5.83.2.1	PayloadFile	48
5.83.2.2	~PayloadFile	48
5.84	DREService::PerlProcessor Class Reference	48
5.84.1	Constructor & Destructor Documentation	49
5.84.1.1	PerlProcessor	49
5.84.1.2	~PerlProcessor	49
5.85	Permission Class Reference	49
5.86	PermissionGACL Class Reference	49
5.87	Policy Class Reference	50
5.88	ArcSec::Charon::PolicyLocation Class Reference	50
5.89	ContinuationPlugins::result_t Class Reference	50
5.90	RunFunction Class Reference	50
5.91	RunParallel Class Reference	50

5.92 gridftpd::RunPlugin Class Reference	50
5.93 RunPlugin Class Reference	51
5.94 RunPlugins Class Reference	51
5.95 RunRedirected Class Reference	51
5.96 Server Class Reference	51
5.97 FileRoot::ServerParams Class Reference	51
5.98 ArcSec::Service_AA Class Reference	52
5.98.1 Detailed Description	52
5.99 Arc::Service_JavaWrapper Class Reference	52
5.99.1 Member Function Documentation	52
5.99.1.1 process	52
5.100 Arc::Service_PythonWrapper Class Reference	52
5.100.1 Member Function Documentation	52
5.100.1.1 process	52
5.101 ArcSec::Service_SLCS Class Reference	53
5.101.1 Detailed Description	53
5.102 SPService::Service_SP Class Reference	53
5.102.1 Detailed Description	53
5.102.2 Constructor & Destructor Documentation	53
5.102.2.1 Service_SP	54
5.102.3 Member Function Documentation	54
5.102.3.1 process	54
5.103 SimpleMap Class Reference	54
5.104 UnixMap::source_t Struct Reference	54
5.105 AuthUser::source_t Struct Reference	54
5.106 DREService::Task Class Reference	54
5.106.1 Constructor & Destructor Documentation	55
5.106.1.1 Task	55
5.106.1.2 ~Task	55
5.107 DREService::TaskQueue Class Reference	55
5.107.1 Constructor & Destructor Documentation	55
5.107.1.1 TaskQueue	55
5.107.1.2 ~TaskQueue	55
5.107.2 Member Function Documentation	55

5.107.2.1 pushTask	55
5.107.2.2 shiftTask	56
5.108DREService::TaskSet Class Reference	56
5.108.1 Constructor & Destructor Documentation	56
5.108.1.1 TaskSet	56
5.108.1.2 ~TaskSet	56
5.108.2 Member Function Documentation	56
5.108.2.1 removeTask	56
5.109DREService::PerlProcessor::ThreadInterface Struct Reference	56
5.110UnixMap::unix_user_t Class Reference	57
5.111UnixMap Class Reference	57
5.112UrlMapConfig Class Reference	57
5.113gridftpd::UrlMapConfig Class Reference	57
5.114userspec_t Class Reference	57
5.115value_for_shell Class Reference	57
5.116voms Struct Reference	58
5.116.1 Detailed Description	58
5.116.2 Field Documentation	58
5.116.2.1 attrs	58
5.116.2.2 server	58
5.116.2.3 voname	58
5.117voms_attrs Struct Reference	58
5.117.1 Detailed Description	59
5.117.2 Field Documentation	59
5.117.2.1 cap	59
5.117.2.2 group	59
5.117.2.3 role	59
5.118ZeroUInt Class Reference	59
5.118.1 Detailed Description	59

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	16
AuthEvaluator	17
AuthUser	17
AuthVO	17
CacheConfig	17
CacheConfigException	18
Cache::CacheService	19
ArcSec::Charon	21
GridFTP_Commands::close_semaphore_t	21
ContinuationPlugins::command_t	21
CommFIFO	21
ConfigSections	22
gridftpd::ConfigSections	22
ContinuationPlugins	22
ARex::CountedResource	22
gridftpd::Daemon	22
GridFTP_Commands::data_buffer_t	22
DataStaging::DataDeliveryService	23
DirectAccess::diraccess_t	24
DirectAccess	24
DirEntry	24
DREService::DREWebService	24
DTRGenerator	26
DTRInfo	29
CommFIFO::elem_t	30
Entry	30
ARex::FileChunks	30

ARex::FileChunksList	31
ARex::FileChunksRef	32
FileData	32
FileNode	32
FilePlugin	32
DirectFilePlugin	24
GACLPlugin	33
JobPlugin	40
FileRoot	32
gm_dirs_	33
gridftp::GMEEnvironment	33
GMEEnvironment	34
GridFTP_Commands	34
GridFTP_Commands_timeout	35
ARex::GridManager	35
AuthUser::group_t	35
Hopi::Hopi	35
Identity	35
IdentityGACL	36
Index	37
ISIS::ISISService	37
ISIS::ISISSecAttr	37
Identity::Item	38
IdentityItemDN	36
IdentityItemVOMS	36
ObjectAccess::Item	37
Janitor	38
job_state_rec_t	39
JobDescription	39
JobLocalDescription	40
JobLog	40
ARex::JobRecord	41
JobsList	41
JobsListConfig	41
JobUser	41
JobUserHelper	41
JobUsers	42
gridftp::LdapQuery	42
gridftp::LdapQueryError	43
RunPlugin::lib_plugin_t	44
gridftp::RunPlugin::lib_plugin_t	44
ARex::LoggerClient	44
LRMSResult	44
ISIS::Neighbor_Container	44
numvalue_for_shell	44
ObjectAccess	45
ObjectAccessGACL	45
ARex::OptimizedInformationContainer	45

gridftpd::ParallelLdapQueries	45
ARex::PayloadBigFile	46
Hopi::PayloadBigFile	46
ARex::PayloadFAFile	47
Hopi::PayloadFile	47
ARex::PayloadFile	48
DREService::PerlProcessor	48
Permission	49
PermissionGACL	49
Policy	50
ArcSec::Charon::PolicyLocation	50
ContinuationPlugins::result_t	50
RunFunction	50
RunParallel	50
gridftpd::RunPlugin	50
RunPlugin	51
RunPlugins	51
RunRedirected	51
Server	51
FileRoot::ServerParams	51
ArcSec::Service_AA	52
Arc::Service_JavaWrapper	52
Arc::Service_PythonWrapper	52
ArcSec::Service_SLCS	53
SPService::Service_SP	53
SimpleMap	54
UnixMap::source_t	54
AuthUser::source_t	54
DREService::Task	54
DREService::TaskQueue	55
DREService::TaskSet	56
DREService::PerlProcessor::ThreadInterface	56
UnixMap::unix_user_t	57
UnixMap	57
UrlMapConfig	57
gridftpd::UrlMapConfig	57
userspec_t	57
value_for_shell	57
voms	58
voms_attrs	58
ZeroUInt	59

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	16
AuthEvaluator	17
AuthUser	17
AuthVO	17
CacheConfig	17
CacheConfigException	18
Cache::CacheService	19
ArcSec::Charon	21
GridFTP_Commands::close_semaphore_t	21
ContinuationPlugins::command_t	21
CommFIFO	21
ConfigSections	22
gridftpd::ConfigSections	22
ContinuationPlugins	22
ARex::CountedResource	22
gridftpd::Daemon	22
GridFTP_Commands::data_buffer_t	22
DataStaging::DataDeliveryService	
Service for the Delivery layer of data staging	23
DirectAccess::diraccess_t	24
DirectAccess	24
DirectFilePlugin	24
DirEntry	24
DREService::DREWebService	24
DTRGenerator	26
DTRInfo	29
CommFIFO::elem_t	30

Entry	30
ARex::FileChunks	
Representation of delivered file chunks	30
ARex::FileChunksList	
Container for FileChunks instances	31
ARex::FileChunksRef	32
FileData	32
FileNode	32
FilePlugin	32
FileRoot	32
GACLPlugin	33
gm_dirs_	33
gridftp::GMEEnvironment	33
GMEEnvironment	34
GridFTP_Commands	34
GridFTP_Commands_timeout	35
ARex::GridManager	35
AuthUser::group_t	35
Hopi::Hopi	35
Identity	35
IdentityGACL	36
IdentityItemDN	36
IdentityItemVOMS	36
Index	37
ISIS::ISIService	37
ISIS::ISISSecAttr	37
ObjectAccess::Item	37
Identity::Item	38
Janitor	
Class to communicate with Janitor - Dynmaic Runtime Environment	
handler	38
job_state_rec_t	39
JobDescription	39
JobLocalDescription	40
JobLog	40
JobPlugin	40
ARex::JobRecord	41
JobsList	41
JobsListConfig	41
JobUser	41
JobUserHelper	41
JobUsers	42
gridftp::LdapQuery	42
gridftp::LdapQueryError	43
RunPlugin::lib_plugin_t	44
gridftp::RunPlugin::lib_plugin_t	44
ARex::LoggerClient	44
LRMSResult	44
ISIS::Neighbor_Container	44
numvalue_for_shell	44

ObjectAccess	45
ObjectAccessGACL	45
ARex::OptimizedInformationContainer	45
gridftpd::ParallelLdapQueries	45
ARex::PayloadBigFile	46
Hopi::PayloadBigFile	46
ARex::PayloadFAFile	47
Hopi::PayloadFile	47
ARex::PayloadFile	48
DREService::PerlProcessor	48
Permission	49
PermissionGACL	49
Policy	50
ArcSec::Charon::PolicyLocation	50
ContinuationPlugins::result_t	50
RunFunction	50
RunParallel	50
gridftpd::RunPlugin	50
RunPlugin	51
RunPlugins	51
RunRedirected	51
Server	51
FileRoot::ServerParams	51
ArcSec::Service_AA	52
Arc::Service_JavaWrapper	52
Arc::Service_PythonWrapper	52
ArcSec::Service_SLCS	53
SPService::Service_SP	53
SimpleMap	54
UnixMap::source_t	54
AuthUser::source_t	54
DREService::Task	54
DREService::TaskQueue	55
DREService::TaskSet	56
DREService::PerlProcessor::ThreadInterface	56
UnixMap::unix_user_t	57
UnixMap	57
UrlMapConfig	57
gridftpd::UrlMapConfig	57
userspec_t	57
value_for_shell	57
voms	58
voms_attrs	58
ZeroUInt	59

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, 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 [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, 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::FileAccess* ARex::ARexJob::CreateFile (const std::string & filename)`

Creates file in job's session directory and returns handler

5.2.3.5 `bool ARex::ARexJob::Failed (void)`

Returns true if job has failed

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

Returns textual description of failure of last operation

5.2.3.7 `bool ARex::ARexJob::GetDescription (Arc::XMLNode & jsdl)`

Fills provided jsdl with job description

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

Return ID assigned to job

5.2.3.9 `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.

5.2.3.10 `std::string ARex::ARexJob::LogDir (void)`

Returns name of virtual log directory

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

Returns list of existing log files

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

Opens directory inside session directory

5.2.3.13 `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

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

Opens log file in control directory

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

Resume execution of job after error

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

Returns path to session directory

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

Returns current state of job

5.2.3.18 `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

5.2.3.19 `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.

5.2.3.20 `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)
- 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)

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 ConfigSections Class Reference

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

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

5.15 gridftpdc::ConfigSections Class Reference

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

- gridftpdc/conf/conf_sections.h

5.16 ContinuationPlugins Class Reference

Data Structures

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

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

- plugins.h

5.17 ARex::CountedResource Class Reference

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

- arex.h

5.18 gridftpdc::Daemon Class Reference

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

- daemon.h

5.19 GridFTP_Commands::data_buffer_t Struct Reference

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

- commands.h

5.20 DataStaging::DataDeliveryService Class Reference

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

Public Member Functions

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

5.20.1 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

5.21 DirectAccess::diraccess_t Struct Reference

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

- `fileplugin.h`

5.22 DirectAccess Class Reference

Data Structures

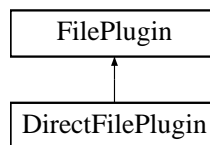
- struct [diraccess_t](#)

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

- `fileplugin.h`

5.23 DirectFilePlugin Class Reference

Inheritance diagram for DirectFilePlugin:



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

- `fileplugin.h`

5.24 DirEntry Class Reference

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

- `fileroot.h`

5.25 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](#)

5.25.1 Constructor & Destructor Documentation

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

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

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

Destructor.

5.25.2 Member Function Documentation

5.25.2.1 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

5.25.2.2 `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>outmsg</i>	outgoing message

Returns

Status of the result achieved

5.25.3 Field Documentation

5.25.3.1 `Arc::Logger DREService::DREWebService::logger [static, protected]`

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

5.25.3.2 `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

5.26 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 &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)

5.26.1 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.

5.26.2 Constructor & Destructor Documentation

5.26.2.1 `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

5.26.2.2 `DTRGenerator::~~DTRGenerator ()`

Stop Generator

5.26.3 Member Function Documentation

5.26.3.1 `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
------------	----------------------------------

5.26.3.2 `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

5.26.3.3 `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.

5.26.3.4 `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.

5.26.3.5 `virtual void DTRGenerator::receivedDTR (DataStaging::DTR & 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
------------	---

5.26.3.6 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.
------------	-------------------------

5.26.3.7 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

5.27 DTRInfo Class Reference

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

Public Member Functions

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

5.27.1 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.

5.27.2 Constructor & Destructor Documentation

5.27.2.1 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

5.28 CommFIFO::elem_t Class Reference

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

- commfifo.h

5.29 Entry Class Reference

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

- Entry.h

5.30 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)

5.30.1 Detailed Description

Representation of delivered file chunks.

5.30.2 Member Function Documentation

5.30.2.1 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.

5.30.2.2 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](#)

5.31 ARex::FileChunksList Class Reference

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

Public Member Functions

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

5.31.1 Detailed Description

Container for [FileChunks](#) instances.

5.31.2 Member Function Documentation

5.31.2.1 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](#)

5.32 ARex::FileChunksRef Class Reference

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

- FileChunks.h

5.33 FileData Class Reference

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

- info_types.h

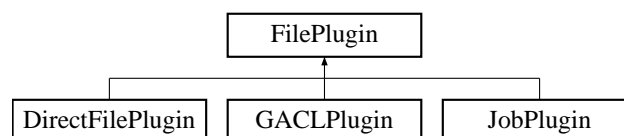
5.34 FileNode Class Reference

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

- fileroot.h

5.35 FilePlugin Class Reference

Inheritance diagram for FilePlugin:



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

- fileroot.h

5.36 FileRoot Class Reference

Data Structures

- class [ServerParams](#)

Friends

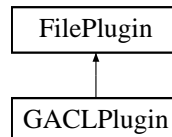
- class **GridFTP_Commands**

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

- fileroot.h

5.37 GACLPlugin Class Reference

Inheritance diagram for GACLPlugin:



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

- gacplugin.h

5.38 gm_dirs_ Struct Reference

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

- jobplugin.h

5.39 gridftp::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

5.39.1 Member Function Documentation

5.39.1.1 std::string gridftp::GMEEnvironment::nordugrid_config_loc (void) const

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

5.39.1.2 `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:

- `gridftp/conf/environment.h`

5.40 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`

5.40.1 Member Function Documentation

5.40.1.1 `std::string GMEEnvironment::nordugrid_config_loc (void) const`

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

5.40.1.2 `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`

5.41 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

5.42 GridFTP_Commands_timeout Class Reference

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

- commands.h

5.43 ARex::GridManager Class Reference

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

- grid_manager.h

5.44 AuthUser::group_t Class Reference

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

- auth.h

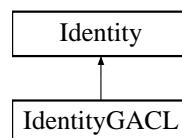
5.45 Hopi::Hopi Class Reference

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

- hopi.h

5.46 Identity Class Reference

Inheritance diagram for Identity:



Data Structures

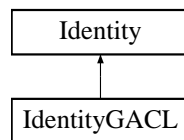
- class [Item](#)

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

- `identity.h`

5.47 IdentityGACL Class Reference

Inheritance diagram for IdentityGACL:

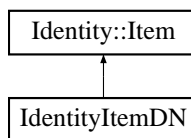


The documentation for this class was generated from the following file:

- `identity_gacl.h`

5.48 IdentityItemDN Class Reference

Inheritance diagram for IdentityItemDN:

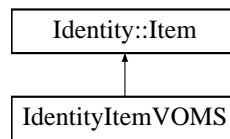


The documentation for this class was generated from the following file:

- `identity_dn.h`

5.49 IdentityItemVOMS Class Reference

Inheritance diagram for IdentityItemVOMS:



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

- identity_voms.h

5.50 Index Class Reference

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

- Index.h

5.51 ISIS::ISIService Class Reference

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

- isis.h

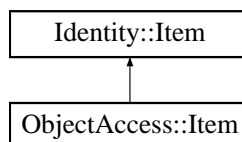
5.52 ISIS::ISISSecAttr Class Reference

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

- security.h

5.53 ObjectAccess::Item Class Reference

Inheritance diagram for ObjectAccess::Item:

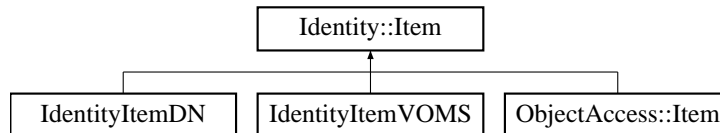


The documentation for this class was generated from the following file:

- object_access.h

5.54 Identity::Item Class Reference

Inheritance diagram for Identity::Item:



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

- identity.h

5.55 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)

5.55.1 Detailed Description

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

5.55.2 Constructor & Destructor Documentation

5.55.2.1 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.

5.55.3 Member Function Documentation

5.55.3.1 `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.

5.55.3.2 `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.

5.55.3.3 `Result Janitor::result (void)`

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

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

5.55.3.4 `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`

5.56 `job_state_rec_t` Struct Reference

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

- `grid-manager/jobs/job.h`

5.57 JobDescription Class Reference

Friends

- class **JobsList**

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

- grid-manager/jobs/job.h

5.58 JobLocalDescription Class Reference

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

- info_types.h

5.59 JobLog Class Reference

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

5.59.1 Detailed Description

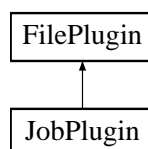
Put short information into log when every job starts/finishes. And store more detailed information for Reporter.

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

- job_log.h

5.60 JobPlugin Class Reference

Inheritance diagram for JobPlugin:



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

- jobplugin.h

5.61 ARex::JobRecord Class Reference

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

- JobRecord.h

5.62 JobsList Class Reference

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

- states.h

5.63 JobsListConfig Class Reference

```
#include <states.h>
```

Friends

- class **JobsList**

5.63.1 Detailed Description

Class to represent information read from configuration.

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

- states.h

5.64 JobUser Class Reference

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

- users.h

5.65 JobUserHelper Class Reference

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

- users.h

5.66 JobUsers Class Reference

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

- `users.h`

5.67 gridftpdp::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](#) ()

5.67.1 Detailed Description

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

5.67.2 Member Enumeration Documentation

5.67.2.1 enum gridftpdp::LdapQuery::Scope

Scope for a LDAP queries. Use when querying.

5.67.3 Constructor & Destructor Documentation

5.67.3.1 gridftpdp::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](#).

5.67.3.2 gridftpdp::LdapQuery::~~LdapQuery ()

Destructor. Will disconnect from the ldapserver if still connected.

5.67.4 Member Function Documentation

5.67.4.1 std::string gridftpdp::LdapQuery::Host ()

Returns the hostname of the ldap-server.

5.67.4.2 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.

5.67.4.3 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

5.68 gridftpdp::LdapQueryError Class Reference

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

Public Member Functions

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

5.68.1 Detailed Description

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

5.68.2 Constructor & Destructor Documentation

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

Standard exception class constructor.

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

- ldapquery.h

5.69 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

5.70 gridftpd::RunPlugin::lib_plugin_t Union Reference

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

- gridftpd/run/run_plugin.h

5.71 ARex::LoggerClient Class Reference

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

- client.h

5.72 LRMSResult Class Reference

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

- info_types.h

5.73 ISIS::Neighbor_Container Class Reference

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

- isis.h

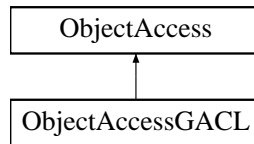
5.74 numvalue_for_shell Class Reference

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

- job_desc.h

5.75 ObjectAccess Class Reference

Inheritance diagram for ObjectAccess:



Data Structures

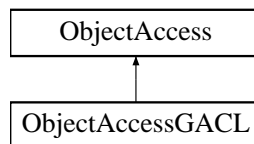
- class [Item](#)

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

- `object_access.h`

5.76 ObjectAccessGACL Class Reference

Inheritance diagram for ObjectAccessGACL:



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

- `object_access_gacl.h`

5.77 ARex::OptimizedInformationContainer Class Reference

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

- `arex.h`

5.78 gridftp::ParallelLdapQueries Class Reference

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

5.78.1 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

5.79 ARex::PayloadBigFile Class Reference

Public Member Functions

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

5.79.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

5.79.1.2 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

5.80 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

5.80.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

5.80.1.2 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

5.81 ARex::PayloadFAFile Class Reference

Public Member Functions

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

5.81.1 Constructor & Destructor Documentation

5.81.1.1 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

5.82 Hopi::PayloadFile Class Reference

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

Public Member Functions

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

5.82.1 Detailed Description

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

5.82.2 Constructor & Destructor Documentation

5.82.2.1 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.

5.82.2.2 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

5.83 ARex::PayloadFile Class Reference

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

Public Member Functions

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

5.83.1 Detailed Description

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

5.83.2 Constructor & Destructor Documentation

5.83.2.1 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.

5.83.2.2 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

5.84 DREService::PerlProcessor Class Reference

Data Structures

- struct [ThreadInterface](#)

Public Member Functions

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

5.84.1 Constructor & Destructor Documentation

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

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

5.84.1.2 `virtual DREService::PerlProcessor::~~PerlProcessor (void) [virtual]`

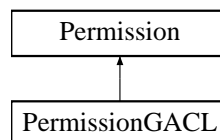
Destructor.

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

- PerlProcessor.h

5.85 Permission Class Reference

Inheritance diagram for Permission:

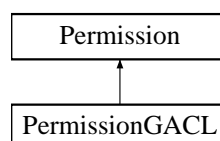


The documentation for this class was generated from the following file:

- permission.h

5.86 PermissionGACL Class Reference

Inheritance diagram for PermissionGACL:



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

- `permission_gacl.h`

5.87 Policy Class Reference

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

- `Policy.h`

5.88 ArcSec::Charon::PolicyLocation Class Reference

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

- `charon.h`

5.89 ContinuationPlugins::result_t Class Reference

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

- `plugins.h`

5.90 RunFunction Class Reference

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

- `run_function.h`

5.91 RunParallel Class Reference

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

- `run_parallel.h`

5.92 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`

5.93 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`

5.94 RunPlugins Class Reference

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

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

5.95 RunRedirected Class Reference

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

- `run_redirected.h`

5.96 Server Class Reference

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

- `Server.h`

5.97 FileRoot::ServerParams Class Reference

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

- `fileroot.h`

5.98 ArcSec::Service_AA Class Reference

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

5.98.1 Detailed Description

A Service which includes the AttributeAuthority functionality; it accepts the <sampl:-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 <sampl:Assertion> which includes the <Attribute>

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

- aaservice.h

5.99 Arc::Service_JavaWrapper Class Reference

Public Member Functions

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

5.99.1 Member Function Documentation

5.99.1.1 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

5.100 Arc::Service_PythonWrapper Class Reference

Public Member Functions

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

5.100.1 Member Function Documentation

5.100.1.1 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

5.101 ArcSec::Service_SLCS Class Reference

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

5.101.1 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

5.102 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 &)

5.102.1 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

5.102.2 Constructor & Destructor Documentation

5.102.2.1 SPSERVICE::Service_SP::Service_SP (Arc::Config * *cfg*)

Constructor

5.102.3 Member Function Documentation

5.102.3.1 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

5.103 SimpleMap Class Reference

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

- simplemap.h

5.104 UnixMap::source_t Struct Reference

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

- unixmap.h

5.105 AuthUser::source_t Struct Reference

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

- auth.h

5.106 DRESERVICE::Task Class Reference

Public Member Functions

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

5.106.1 Constructor & Destructor Documentation

5.106.1.1 DREService::Task::Task (int *taskId*, Arc::Message * *request*, Arc::Message * *response*)

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

5.106.1.2 virtual DREService::Task::~~Task (void) [virtual]

Destructor.

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

- Task.h

5.107 DREService::TaskQueue Class Reference

Public Member Functions

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

5.107.1 Constructor & Destructor Documentation

5.107.1.1 DREService::TaskQueue::TaskQueue (int *length*)

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

5.107.1.2 virtual DREService::TaskQueue::~~TaskQueue (void) [virtual]

Destructor.

5.107.2 Member Function Documentation

5.107.2.1 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.

5.107.2.2 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

5.108 DREService::TaskSet Class Reference

Public Member Functions

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

5.108.1 Constructor & Destructor Documentation

5.108.1.1 DREService::TaskSet::TaskSet (int size)

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

5.108.1.2 virtual DREService::TaskSet::~~TaskSet (void) [virtual]

Destructor.

5.108.2 Member Function Documentation

5.108.2.1 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, themethod 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

5.109 DREService::PerlProcessor::ThreadInterface Struct Reference

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

- PerlProcessor.h

5.110 UnixMap::unix_user_t Class Reference

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

- unixmap.h

5.111 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

5.112 UrlMapConfig Class Reference

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

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

5.113 gridftpd::UrlMapConfig Class Reference

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

- gridftpd/conf/conf_map.h

5.114 userspec_t Class Reference

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

- userspec.h

5.115 value_for_shell Class Reference

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

- job_desc.h

5.116 voms Struct Reference

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

Data Fields

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

5.116.1 Detailed Description

VOMS data

5.116.2 Field Documentation

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

User's characteristics

5.116.2.2 std::string voms::server

The VOMS server DN, as from its certificate

5.116.2.3 std::string 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](#)

5.117 voms_attrs Struct Reference

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

Data Fields

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

5.117.1 Detailed Description

VOMS attributes

5.117.2 Field Documentation

5.117.2.1 `std::string voms_attrs::cap`

user's capability

5.117.2.2 `std::string voms_attrs::group`

user's group

5.117.2.3 `std::string voms_attrs::role`

user's role

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

- `auth.h`

5.118 ZeroUInt Class Reference

```
#include <states.h>
```

5.118.1 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:

- `states.h`