

Hosting Environment (Daemon) Services

Generated by Doxygen 1.7.3

Wed Mar 23 2011 15:20:14

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	CommFIFO Class Reference	21
5.12	ConfigSections Class Reference	21
5.13	gridftpd::ConfigSections Class Reference	21
5.14	ContinuationPlugins Class Reference	22
5.15	ARex::CountedResource Class Reference	22
5.16	gridftpd::Daemon Class Reference	22
5.17	DirectAccess::diraccess_t Struct Reference	22
5.18	DirectAccess Class Reference	22
5.19	DirectFilePlugin Class Reference	23
5.20	DirEntry Class Reference	23
5.21	DREService::DREWebService Class Reference	23
5.21.1	Constructor & Destructor Documentation	24
5.21.1.1	DREWebService	24
5.21.1.2	~DREWebService	24
5.21.2	Member Function Documentation	24
5.21.2.1	makeFault	24
5.21.2.2	process	24
5.21.3	Field Documentation	24
5.21.3.1	logger	24
5.21.3.2	ns_	25

5.22	DTRGenerator Class Reference	25
5.22.1	Detailed Description	25
5.22.2	Constructor & Destructor Documentation	25
5.22.2.1	DTRGenerator	25
5.22.2.2	~DTRGenerator	25
5.22.3	Member Function Documentation	26
5.22.3.1	cancelJob	26
5.22.3.2	checkUploadedFiles	26
5.22.3.3	queryJobFinished	26
5.22.3.4	receiveDTR	26
5.22.3.5	receiveJob	27
5.23	DTRInfo Class Reference	27
5.23.1	Detailed Description	27
5.23.2	Constructor & Destructor Documentation	27
5.23.2.1	DTRInfo	27
5.24	Entry Class Reference	27
5.25	ARex::FileChunks Class Reference	28
5.25.1	Detailed Description	28
5.25.2	Member Function Documentation	28
5.25.2.1	Release	28
5.25.2.2	Remove	28
5.26	ARex::FileChunksList Class Reference	28
5.26.1	Detailed Description	29
5.26.2	Member Function Documentation	29
5.26.2.1	Get	29
5.26.2.2	GetStuck	29
5.27	FileData Class Reference	29
5.28	FileNode Class Reference	29
5.29	FilePlugin Class Reference	30
5.30	FileRoot Class Reference	30
5.31	GACLPlugin Class Reference	30
5.32	gm_dirs_ Struct Reference	31
5.33	gridftp::GMEEnvironment Class Reference	31
5.33.1	Member Function Documentation	31
5.33.1.1	nordugrid_config_loc	31
5.33.1.2	support_mail_address	31
5.34	GMEEnvironment Class Reference	31
5.34.1	Member Function Documentation	31
5.34.1.1	nordugrid_config_loc	31
5.34.1.2	support_mail_address	32
5.35	GridFTP_Commands Class Reference	32
5.36	GridFTP_Commands_timeout Class Reference	32
5.37	ARex::GridManager Class Reference	32
5.38	Hopi::Hopi Class Reference	32
5.39	Identity Class Reference	33
5.40	IdentityGACL Class Reference	33
5.41	IdentityItemDN Class Reference	33
5.42	IdentityItemVOMS Class Reference	34
5.43	Index Class Reference	34
5.44	ISIS::ISIService Class Reference	34

5.45	ISIS::ISISSecAttr Class Reference	34
5.46	ObjectAccess::Item Class Reference	34
5.47	Identity::Item Class Reference	35
5.48	Janitor Class Reference	35
5.48.1	Detailed Description	36
5.48.2	Constructor & Destructor Documentation	36
5.48.2.1	Janitor	36
5.48.3	Member Function Documentation	36
5.48.3.1	deploy	36
5.48.3.2	remove	36
5.48.3.3	result	36
5.48.3.4	wait	36
5.49	job_state_rec_t Struct Reference	37
5.50	JobDescription Class Reference	37
5.51	JobLocalDescription Class Reference	37
5.52	JobLog Class Reference	37
5.52.1	Detailed Description	37
5.53	JobPlugin Class Reference	37
5.54	ARex::JobRecord Class Reference	38
5.55	JobsList Class Reference	38
5.56	JobsListConfig Class Reference	38
5.56.1	Detailed Description	38
5.57	JobUser Class Reference	38
5.58	JobUserHelper Class Reference	39
5.59	JobUsers Class Reference	39
5.60	gridftpd::LdapQuery Class Reference	39
5.60.1	Detailed Description	39
5.60.2	Member Enumeration Documentation	39
5.60.2.1	Scope	39
5.60.3	Constructor & Destructor Documentation	40
5.60.3.1	LdapQuery	40
5.60.3.2	~LdapQuery	40
5.60.4	Member Function Documentation	40
5.60.4.1	Host	40
5.60.4.2	Query	40
5.60.4.3	Result	40
5.61	gridftpd::LdapQueryError Class Reference	40
5.61.1	Detailed Description	41
5.61.2	Constructor & Destructor Documentation	41
5.61.2.1	LdapQueryError	41
5.62	RunPlugin::lib_plugin_t Union Reference	41
5.63	gridftpd::RunPlugin::lib_plugin_t Union Reference	41
5.64	ARex::LoggerClient Class Reference	41
5.65	LRMSResult Class Reference	41
5.66	ISIS::Neighbor_Container Class Reference	41
5.67	numvalue_for_shell Class Reference	42
5.68	ObjectAccess Class Reference	42
5.69	ObjectAccessGACL Class Reference	42
5.70	ARex::OptimizedInformationContainer Class Reference	42
5.71	gridftpd::ParallelLdapQueries Class Reference	43

5.71.1 Detailed Description	43
5.72 Hopi::PayloadBigFile Class Reference	43
5.72.1 Constructor & Destructor Documentation	43
5.72.1.1 PayloadBigFile	43
5.72.1.2 ~PayloadBigFile	43
5.73 ARex::PayloadBigFile Class Reference	43
5.73.1 Constructor & Destructor Documentation	44
5.73.1.1 PayloadBigFile	44
5.73.1.2 ~PayloadBigFile	44
5.74 ARex::PayloadFile Class Reference	44
5.74.1 Detailed Description	44
5.74.2 Constructor & Destructor Documentation	44
5.74.2.1 PayloadFile	44
5.74.2.2 ~PayloadFile	44
5.75 Hopi::PayloadFile Class Reference	45
5.75.1 Detailed Description	45
5.75.2 Constructor & Destructor Documentation	45
5.75.2.1 PayloadFile	45
5.75.2.2 ~PayloadFile	45
5.76 DREService::PerlProcessor Class Reference	45
5.76.1 Constructor & Destructor Documentation	46
5.76.1.1 PerlProcessor	46
5.76.1.2 ~PerlProcessor	46
5.77 Permission Class Reference	46
5.78 PermissionGACL Class Reference	46
5.79 Policy Class Reference	47
5.80 ArcSec::Charon::PolicyLocation Class Reference	47
5.81 ContinuationPlugins::result_t Class Reference	47
5.82 RunFunction Class Reference	47
5.83 RunParallel Class Reference	47
5.84 gridftpd::RunPlugin Class Reference	47
5.85 RunPlugin Class Reference	48
5.86 RunPlugins Class Reference	48
5.87 RunRedirected Class Reference	48
5.88 Server Class Reference	48
5.89 FileRoot::ServerParams Class Reference	48
5.90 ArcSec::Service_AA Class Reference	48
5.90.1 Detailed Description	49
5.91 Arc::Service_JavaWrapper Class Reference	49
5.91.1 Member Function Documentation	49
5.91.1.1 process	49
5.92 Arc::Service_PythonWrapper Class Reference	49
5.92.1 Member Function Documentation	49
5.92.1.1 process	49
5.93 ArcSec::Service_SLCS Class Reference	50
5.93.1 Detailed Description	50
5.94 SPService::Service_SP Class Reference	50
5.94.1 Detailed Description	50
5.94.2 Constructor & Destructor Documentation	50
5.94.2.1 Service_SP	50

5.94.3	Member Function Documentation	51
5.94.3.1	process	51
5.95	SimpleMap Class Reference	51
5.96	DREService::Task Class Reference	51
5.96.1	Constructor & Destructor Documentation	51
5.96.1.1	Task	51
5.96.1.2	~Task	51
5.97	DREService::TaskQueue Class Reference	51
5.97.1	Constructor & Destructor Documentation	52
5.97.1.1	TaskQueue	52
5.97.1.2	~TaskQueue	52
5.97.2	Member Function Documentation	52
5.97.2.1	pushTask	52
5.97.2.2	shiftTask	52
5.98	DREService::TaskSet Class Reference	52
5.98.1	Constructor & Destructor Documentation	52
5.98.1.1	TaskSet	52
5.98.1.2	~TaskSet	53
5.98.2	Member Function Documentation	53
5.98.2.1	removeTask	53
5.99	UnixMap Class Reference	53
5.100	UrlMapConfig Class Reference	53
5.101	gridftpd::UrlMapConfig Class Reference	53
5.102	userspec_t Class Reference	53
5.103	value_for_shell Class Reference	54
5.104	voms Struct Reference	54
5.104.1	Detailed Description	54
5.104.2	Field Documentation	54
5.104.2.1	server	54
5.104.2.2	std	54
5.104.2.3	voname	54
5.105	voms_attrs Struct Reference	54
5.105.1	Detailed Description	55
5.105.2	Field Documentation	55
5.105.2.1	cap	55
5.105.2.2	group	55
5.105.2.3	role	55
5.106	ZeroUInt Class Reference	55
5.106.1	Detailed Description	55

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
CommFIFO	21
ConfigSections	21
gridftpd::ConfigSections	21
ContinuationPlugins	22
ARex::CountedResource	22
gridftpd::Daemon	22
DirectAccess::diraccess_t	22
DirectAccess	22
DirEntry	23
DREService::DREWebService	23
DTRGenerator	25
DTRInfo	27
Entry	27
ARex::FileChunks	28
ARex::FileChunksList	28
FileData	29
FileNode	29
FilePlugin	30
DirectFilePlugin	23

GACLPlugin	30
JobPlugin	37
FileRoot	30
gm_dirs_	31
gridftpd::GMEEnvironment	31
GMEEnvironment	31
GridFTP_Commands	32
GridFTP_Commands_timeout	32
ARex::GridManager	32
Hopi::Hopi	32
Identity	33
IdentityGACL	33
Index	34
ISIS::ISIService	34
ISIS::ISISecAttr	34
Identity::Item	35
IdentityItemDN	33
IdentityItemVOMS	34
ObjectAccess::Item	34
Janitor	35
job_state_rec_t	37
JobDescription	37
JobLocalDescription	37
JobLog	37
ARex::JobRecord	38
JobsList	38
JobsListConfig	38
JobUser	38
JobUserHelper	39
JobUsers	39
gridftpd::LdapQuery	39
gridftpd::LdapQueryError	40
RunPlugin::lib_plugin_t	41
gridftpd::RunPlugin::lib_plugin_t	41
ARex::LoggerClient	41
LRMSResult	41
ISIS::Neighbor_Container	41
numvalue_for_shell	42
ObjectAccess	42
ObjectAccessGACL	42
ARex::OptimizedInformationContainer	42
gridftpd::ParallelLdapQueries	43
Hopi::PayloadBigFile	43
ARex::PayloadBigFile	43
ARex::PayloadFile	44
Hopi::PayloadFile	45
DREService::PerlProcessor	45
Permission	46
PermissionGACL	46

Policy	47
ArcSec::Charon::PolicyLocation	47
ContinuationPlugins::result_t	47
RunFunction	47
RunParallel	47
gridftpd::RunPlugin	47
RunPlugin	48
RunPlugins	48
RunRedirected	48
Server	48
FileRoot::ServerParams	48
ArcSec::Service_AA	48
Arc::Service_JavaWrapper	49
Arc::Service_PythonWrapper	49
ArcSec::Service_SLCS	50
SPService::Service_SP	50
SimpleMap	51
DREService::Task	51
DREService::TaskQueue	51
DREService::TaskSet	52
UnixMap	53
UrlMapConfig	53
gridftpd::UrlMapConfig	53
userspec_t	53
value_for_shell	54
voms	54
voms_attrs	54
ZeroUInt	55

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
CommFIFO	21
ConfigSections	21
gridftpd::ConfigSections	21
ContinuationPlugins	22
ARex::CountedResource	22
gridftpd::Daemon	22
DirectAccess::diraccess_t	22
DirectAccess	22
DirectFilePlugin	23
DirEntry	23
DREService::DREWebService	23
DTRGenerator	25
DTRInfo	27
Entry	27
ARex::FileChunks (Representation of delivered file chunks)	28
ARex::FileChunksList (Container for FileChunks (p. 28) instances)	28
FileData	29
FileNode	29
FilePlugin	30

FileRoot	30
GACLPlugin	30
gm_dirs	31
gridftpd::GMEEnvironment	31
GMEEnvironment	31
GridFTP_Commands	32
GridFTP_Commands_timeout	32
ARex::GridManager	32
Hopi::Hopi	32
Identity	33
IdentityGACL	33
IdentityItemDN	33
IdentityItemVOMS	34
Index	34
ISIS::ISISService	34
ISIS::ISISSecAttr	34
ObjectAccess::Item	34
Identity::Item	35
Janitor (Class to communicate with Janitor (p. 35) - Dynmaic Runtime Environment handler)	35
job_state_rec_t	37
JobDescription	37
JobLocalDescription	37
JobLog	37
JobPlugin	37
ARex::JobRecord	38
JobsList	38
JobsListConfig	38
JobUser	38
JobUserHelper	39
JobUsers	39
gridftpd::LdapQuery	39
gridftpd::LdapQueryError	40
RunPlugin::lib_plugin_t	41
gridftpd::RunPlugin::lib_plugin_t	41
ARex::LoggerClient	41
LRMSResult	41
ISIS::Neighbor_Container	41
numvalue_for_shell	42
ObjectAccess	42
ObjectAccessGACL	42
ARex::OptimizedInformationContainer	42
gridftpd::ParallelLdapQueries	43
Hopi::PayloadBigFile	43
ARex::PayloadBigFile	43
ARex::PayloadFile	44
Hopi::PayloadFile	45
DREService::PerlProcessor	45
Permission	46
PermissionGACL	46

Policy	47
ArcSec::Charon::PolicyLocation	47
ContinuationPlugins::result_t	47
RunFunction	47
RunParallel	47
gridftpd::RunPlugin	47
RunPlugin	48
RunPlugins	48
RunRedirected	48
Server	48
FileRoot::ServerParams	48
ArcSec::Service_AA	48
Arc::Service_JavaWrapper	49
Arc::Service_PythonWrapper	49
ArcSec::Service_SLCS	50
SPService::Service_SP	50
SimpleMap	51
DREService::Task	51
DREService::TaskQueue	51
DREService::TaskSet	52
UnixMap	53
UrlMapConfig	53
gridftpd::UrlMapConfig	53
userspec_t	53
value_for_shell	54
voms	54
voms_attrs	54
ZeroUInt	55

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)
- int **CreateFile** (const std::string &filename)

- int **OpenFile** (const std::string &filename, bool for_read, bool for_write)
- int **OpenLogFile** (const std::string &name)
- Glib::Dir * **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 int 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 Glib::Dir* ARex::ARexJob::OpenDir (const std::string & dirname)

Opens directory inside session directory

5.2.3.13 `int 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 (bool & job_pending)`

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

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

Returns current state of job

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** (p. 17) 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** (p. 17)

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 **JobUser** &user)
- Arc::MCC_Status **CacheLink** (Arc::XMLNode in, Arc::XMLNode out, const **JobUser** &user, const Arc::User &mapped_user)

5.9.1 Detailed Description

CacheService (p. 19) 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** (p. 19). Reads the configuration and determines the validity of the service.

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

Destroy the **CacheService** (p. 19)

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** (p. 19) is valid.

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

Returns true if the **CacheService** (p. 19) 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** (p. 19) is invoked. Directs call to appropriate **CacheService** (p. 19) 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 CommFIFO Class Reference

Data Structures

- class **elem_t**

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

- commfifo.h

5.12 ConfigSections Class Reference

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

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

5.13 gridftpd::ConfigSections Class Reference

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

- gridftpd/conf/conf_sections.h

5.14 ContinuationPlugins Class Reference

Data Structures

- class **command_t**
- class **result_t**

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

- plugins.h

5.15 ARex::CountedResource Class Reference

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

- arex.h

5.16 gridftpd::Daemon Class Reference

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

- daemon.h

5.17 DirectAccess::diraccess_t Struct Reference

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

- fileplugin.h

5.18 DirectAccess Class Reference

Data Structures

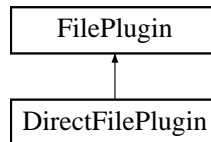
- struct **diraccess_t**

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

- fileplugin.h

5.19 DirectFilePlugin Class Reference

Inheritance diagram for DirectFilePlugin:



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

- fileplugin.h

5.20 DirEntry Class Reference

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

- fileroot.h

5.21 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**

5.21.1 Constructor & Destructor Documentation

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

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

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

Destructor.

5.21.2 Member Function Documentation

5.21.2.1 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

5.21.2.2 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

5.21.3 Field Documentation

5.21.3.1 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.

5.21.3.2 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

5.22 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)
- int **checkUploadedFiles** (**JobDescription** &job)

5.22.1 Detailed Description

A-REX implementation of DTR Generator.

5.22.2 Constructor & Destructor Documentation

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

Start up Generator.

Parameters

<i>user</i>	JobUsers (p. 39) 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

5.22.2.2 DTRGenerator::~~DTRGenerator ()

Stop Generator

5.22.3 Member Function Documentation

5.22.3.1 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
------------	----------------------------------

5.22.3.2 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

5.22.3.3 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** (p. 37) 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.

5.22.3.4 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
------------	---

5.22.3.5 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.
------------	-------------------------

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

- dtr_generator.h

5.23 DTRInfo Class Reference

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

Public Member Functions

- **DTRInfo** (const **JobUsers** &users)

5.23.1 Detailed Description

DTRInfo (p. 27) 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.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 DTRInfo::DTRInfo (const JobUsers & users)

JobUsers (p. 39) is needed to find the correct control dir

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

- dtr_generator.h

5.24 Entry Class Reference

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

- Entry.h

5.25 ARex::FileChunks Class Reference

Representation of delivered file chunks.

```
#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)

5.25.1 Detailed Description

Representation of delivered file chunks.

5.25.2 Member Function Documentation

5.25.2.1 void ARex::FileChunks::Release (void)

Release reference obtained through **FileChunksList::Get()** (p. 29) method. This operation may lead to destruction of FileChunk instance hence previously obtained reference must not be used.

5.25.2.2 void ARex::FileChunks::Remove (void)

Relases reference obtained through Get() method and destroys its instance. Normally this method to be called instead of **Release()** (p. 28) after whole file is delivered in order to free resources associated with **FileChunks** (p. 28) instance.

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

- FileChunks.h

5.26 ARex::FileChunksList Class Reference

Container for **FileChunks** (p. 28) instances.

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

Public Member Functions

- **FileChunks & Get** (std::string path)
- void **Timeout** (int t)
- **FileChunks * GetStuck** (void)
- **FileChunks * GetFirst** (void)

5.26.1 Detailed Description

Container for **FileChunks** (p. 28) instances.

5.26.2 Member Function Documentation

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

Returns previously created **FileChunks** (p. 28) 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.

5.26.2.2 **FileChunks* ARex::FileChunksList::GetStuck (void)**

Returns pointer to first stuck file. File is considered stuck if its Add method was last called more timeout seconds ago.

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

- FileChunks.h

5.27 FileData Class Reference

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

- info_types.h

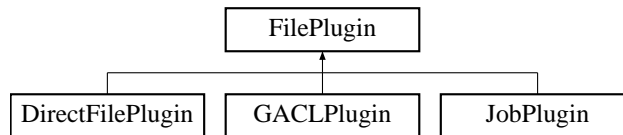
5.28 FileNode Class Reference

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

- fileroot.h

5.29 FilePlugin Class Reference

Inheritance diagram for FilePlugin:



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

- fileroot.h

5.30 FileRoot Class Reference

Data Structures

- class **ServerParams**

Friends

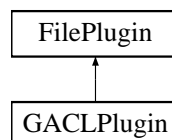
- class **GridFTP_Commands**

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

- fileroot.h

5.31 GACLPlugin Class Reference

Inheritance diagram for GACLPlugin:



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

- gacplugin.h

5.32 gm_dirs_ Struct Reference

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

- jobplugin.h

5.33 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

5.33.1 Member Function Documentation

5.33.1.1 std::string gridftpd::GMEEnvironment::nordugrid_config_loc (void) const

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

5.33.1.2 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

5.34 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

5.34.1 Member Function Documentation

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

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

5.34.1.2 `std::string GEnvironment::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

5.35 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

5.36 GridFTP_Commands_timeout Class Reference

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

- commands.h

5.37 ARex::GridManager Class Reference

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

- grid_manager.h

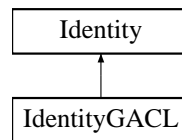
5.38 Hopi::Hopi Class Reference

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

- hopi.h

5.39 Identity Class Reference

Inheritance diagram for Identity:



Data Structures

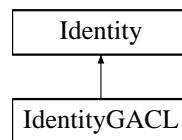
- class **Item**

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

- identity.h

5.40 IdentityGACL Class Reference

Inheritance diagram for IdentityGACL:

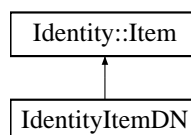


The documentation for this class was generated from the following file:

- identity_gacl.h

5.41 IdentityItemDN Class Reference

Inheritance diagram for IdentityItemDN:

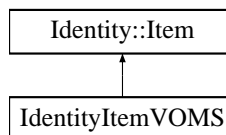


The documentation for this class was generated from the following file:

- identity_dn.h

5.42 IdentityItemVOMS Class Reference

Inheritance diagram for IdentityItemVOMS:



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

- identity_voms.h

5.43 Index Class Reference

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

- Index.h

5.44 ISIS::ISIService Class Reference

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

- isis.h

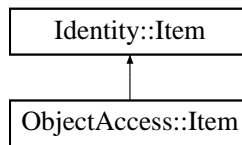
5.45 ISIS::ISISSecAttr Class Reference

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

- security.h

5.46 ObjectAccess::Item Class Reference

Inheritance diagram for ObjectAccess::Item:

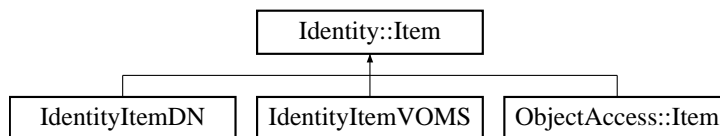


The documentation for this class was generated from the following file:

- object_access.h

5.47 Identity::Item Class Reference

Inheritance diagram for Identity::Item:



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

- identity.h

5.48 Janitor Class Reference

Class to communicate with **Janitor** (p. 35) - Dynmaic Runtime Environment handler.

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

Public Member Functions

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

5.48.1 Detailed Description

Class to communicate with **Janitor** (p. 35) - Dynmaic Runtime Environment handler.

5.48.2 Constructor & Destructor Documentation

5.48.2.1 **Janitor::Janitor** (*const std::string & id*, *const std::string & cdir*, *const GMEEnvironment & env*)

Creates instance representing job entry in **Janitor** (p. 35) database.

Takes id for job identifier and cdir for the control directory of A-Rex. constructor does not register job in the **Janitor** (p. 35). It only associates job with this instance.

5.48.3 Member Function Documentation

5.48.3.1 **bool Janitor::deploy** (*void*)

Registers associated job with **Janitor** (p. 35) and deploys dynamic RTEs.

This operation is asynchronous. Returned true means **Janitor** (p. 35) will be contacted and deployment will start soon. For obtaining result of operation see methods **wait()** (p. 36) and **result()** (p. 36). During this operation janitor utility is called with command register and optionally deploy.

5.48.3.2 **bool Janitor::remove** (*void*)

Removes job from those handled by **Janitor** (p. 35) and releases associated RTEs.

This operation is asynchronous. Returned true means **Janitor** (p. 35) will be contacted and removal will start soon. For obtaining result of operation see methods **wait()** (p. 36) and **result()** (p. 36). During this operation janitor utility is called with command remove.

5.48.3.3 **Result Janitor::result** (*void*)

Returns true if operation initiated by **deploy()** (p. 36) or **remove()** (p. 36) succeeded.

It should be called after **wait()** (p. 36) returned true.

5.48.3.4 **bool Janitor::wait** (*int timeout*)

Wait till operation initiated by **deploy()** (p. 36) or **remove()** (p. 36) 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`

5.49 `job_state_rec_t` Struct Reference

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

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

5.50 JobDescription Class Reference

Friends

- class `JobsList`

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

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

5.51 JobLocalDescription Class Reference

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

- `info_types.h`

5.52 JobLog Class Reference

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

5.52.1 Detailed Description

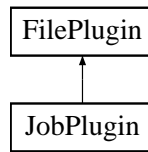
Put short information into log when every job starts/finishes. And store more detailed information for Reporter.

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

- `job_log.h`

5.53 JobPlugin Class Reference

Inheritance diagram for JobPlugin:



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

- jobplugin.h

5.54 ARex::JobRecord Class Reference

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

- JobRecord.h

5.55 JobsList Class Reference

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

- states.h

5.56 JobsListConfig Class Reference

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

Friends

- class **JobsList**

5.56.1 Detailed Description

Class to represent information read from configuration.

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

- states.h

5.57 JobUser Class Reference

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

- users.h

5.58 JobUserHelper Class Reference

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

- users.h

5.59 JobUsers Class Reference

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

- users.h

5.60 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** ()

5.60.1 Detailed Description

LdapQuery (p. 39) class; querying of LDAP servers.

5.60.2 Member Enumeration Documentation

5.60.2.1 enum gridftp::LdapQuery::Scope

Scope for a LDAP queries. Use when querying.

5.60.3 Constructor & Destructor Documentation

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

Constructs a new **LdapQuery** (p. 39) object and sets connection options. The connection is first established when calling `Query`.

5.60.3.2 `gridftp::LdapQuery::~~LdapQuery ()`

Destructor. Will disconnect from the ldapserver if still connected.

5.60.4 Member Function Documentation

5.60.4.1 `std::string gridftp::LdapQuery::Host ()`

Returns the hostname of the ldap-server.

5.60.4.2 `void gridftp::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.

5.60.4.3 `void gridftp::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`

5.61 gridftp::LdapQueryError Class Reference

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

Public Member Functions

- **LdapQueryError** (std::string message)

5.61.1 Detailed Description

LdapQuery (p. 39) exception. Gets thrown when an error occurs in a query.

5.61.2 Constructor & Destructor Documentation

5.61.2.1 gridftpd::LdapQueryError::LdapQueryError (std::string *message*) [inline]

Standard exception class constructor.

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

- ldapquery.h

5.62 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

5.63 gridftpd::RunPlugin::lib_plugin_t Union Reference

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

- gridftpd/run/run_plugin.h

5.64 ARex::LoggerClient Class Reference

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

- client.h

5.65 LRMSResult Class Reference

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

- info_types.h

5.66 ISIS::Neighbor_Container Class Reference

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

- isis.h

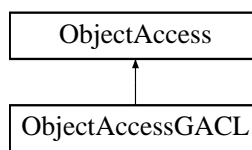
5.67 numvalue_for_shell Class Reference

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

- job_desc.h

5.68 ObjectAccess Class Reference

Inheritance diagram for ObjectAccess:



Data Structures

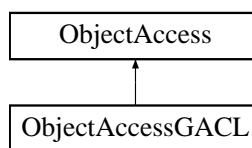
- class **Item**

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

- object_access.h

5.69 ObjectAccessGACL Class Reference

Inheritance diagram for ObjectAccessGACL:



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

- object_access_gacl.h

5.70 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

5.71 gridftp2::ParallelLdapQueries Class Reference

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

5.71.1 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

5.72 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

5.72.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

5.72.1.2 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

5.73 ARex::PayloadBigFile Class Reference

Public Member Functions

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

5.73.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

5.73.1.2 `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

5.74 ARex::PayloadFile Class Reference

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

Public Member Functions

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

5.74.1 Detailed Description

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

5.74.2 Constructor & Destructor Documentation

5.74.2.1 `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.

5.74.2.2 `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

5.75 Hopi::PayloadFile Class Reference

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

Public Member Functions

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

5.75.1 Detailed Description

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

5.75.2 Constructor & Destructor Documentation

5.75.2.1 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.

5.75.2.2 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

5.76 DREService::PerlProcessor Class Reference

Data Structures

- struct **ThreadInterface**

Public Member Functions

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

5.76.1 Constructor & Destructor Documentation

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

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

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

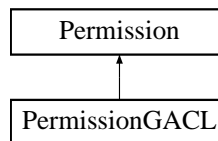
Destructor.

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

- PerlProcessor.h

5.77 Permission Class Reference

Inheritance diagram for Permission:

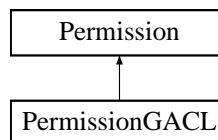


The documentation for this class was generated from the following file:

- permission.h

5.78 PermissionGACL Class Reference

Inheritance diagram for PermissionGACL:



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

- permission_gacl.h

5.79 Policy Class Reference

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

- Policy.h

5.80 ArcSec::Charon::PolicyLocation Class Reference

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

- charon.h

5.81 ContinuationPlugins::result_t Class Reference

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

- plugins.h

5.82 RunFunction Class Reference

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

- run_function.h

5.83 RunParallel Class Reference

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

- run_parallel.h

5.84 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

5.85 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`

5.86 RunPlugins Class Reference

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

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

5.87 RunRedirected Class Reference

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

- `run_redirected.h`

5.88 Server Class Reference

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

- `Server.h`

5.89 FileRoot::ServerParams Class Reference

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

- `fileroot.h`

5.90 ArcSec::Service_AA Class Reference

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


5.90.1 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

5.91 Arc::Service_JavaWrapper Class Reference

Public Member Functions

- virtual Arc::MCC_Status **process** (Arc::Message &, Arc::Message &)

5.91.1 Member Function Documentation

- 5.91.1.1 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

5.92 Arc::Service_PythonWrapper Class Reference

Public Member Functions

- virtual Arc::MCC_Status **process** (Arc::Message &, Arc::Message &)

5.92.1 Member Function Documentation

- 5.92.1.1 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

5.93 ArcSec::Service_SLCS Class Reference

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

5.93.1 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

5.94 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 &)

5.94.1 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

5.94.2 Constructor & Destructor Documentation

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

Constructor

5.94.3 Member Function Documentation

5.94.3.1 `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

5.95 SimpleMap Class Reference

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

- simplemap.h

5.96 DREService::Task Class Reference

Public Member Functions

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

5.96.1 Constructor & Destructor Documentation

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

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

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

Destructor.

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

- Task.h

5.97 DREService::TaskQueue Class Reference

Public Member Functions

- **TaskQueue** (int length)

- virtual `~TaskQueue` (void)
- int `pushTask` (Task *task)
- Task * `shiftTask` ()

5.97.1 Constructor & Destructor Documentation

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

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

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

Destructor.

5.97.2 Member Function Documentation

5.97.2.1 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.

5.97.2.2 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

5.98 DREService::TaskSet Class Reference

Public Member Functions

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

5.98.1 Constructor & Destructor Documentation

5.98.1.1 DREService::TaskSet::TaskSet (int *size*)

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

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

Destructor.

5.98.2 Member Function Documentation

5.98.2.1 `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

5.99 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

5.100 UrlMapConfig Class Reference

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

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

5.101 gridftpd::UrlMapConfig Class Reference

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

- gridftpd/conf/conf_map.h

5.102 userspec_t Class Reference

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

- userspec.h

5.103 value_for_shell Class Reference

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

- job_desc.h

5.104 voms Struct Reference

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

Data Fields

- std::string **server**
- std::string **voname**
- std::vector< **voms_attrs** > **std**

5.104.1 Detailed Description

VOMS data

5.104.2 Field Documentation

5.104.2.1 std::string voms::server

The VOMS server DN, as from its certificate

5.104.2.2 std::vector<voms_attrs> voms::std

User's characteristics

5.104.2.3 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

5.105 voms_attrs Struct Reference

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

Data Fields

- `std::string group`
- `std::string role`
- `std::string cap`

5.105.1 Detailed Description

VOMS attributes

5.105.2 Field Documentation

5.105.2.1 `std::string voms_attrs::cap`

user's capability

5.105.2.2 `std::string voms_attrs::group`

user's group

5.105.2.3 `std::string voms_attrs::role`

user's role

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

- `auth.h`

5.106 ZeroUInt Class Reference

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

5.106.1 Detailed Description

ZeroUInt (p. 55) 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`

Index

- ~CacheService
 - Cache::CacheService, 19
- ~DREWebService
 - DREService::DREWebService, 24
- ~DTRGenerator
 - DTRGenerator, 25
- ~LdapQuery
 - gridftpd::LdapQuery, 40
- ~PayloadBigFile
 - ARex::PayloadBigFile, 44
 - Hopi::PayloadBigFile, 43
- ~PayloadFile
 - ARex::PayloadFile, 44
 - Hopi::PayloadFile, 45
- ~PerlProcessor
 - DREService::PerlProcessor, 46
- ~Task
 - DREService::Task, 51
- ~TaskQueue
 - DREService::TaskQueue, 52
- ~TaskSet
 - DREService::TaskSet, 52
- Arc::Service_JavaWrapper, 49
 - process, 49
- Arc::Service_PythonWrapper, 49
 - process, 49
- ArcSec::Charon, 21
- ArcSec::Charon::PolicyLocation, 47
- ArcSec::Service_AA, 48
- ArcSec::Service_SLCS, 50
- ARex::ARexGMConfig, 13
- ARex::ARexJob, 13
 - ARexJob, 14
 - Cancel, 14
 - ChooseSessionDir, 14
 - Clean, 14
 - CreateFile, 15
 - Failed, 15
 - Failure, 15
 - GetDescription, 15
 - ID, 15
 - Jobs, 15
 - LogDir, 15
 - LogFiles, 15
 - OpenDir, 15
 - OpenFile, 15
 - OpenLogFile, 16
 - Resume, 16
 - SessionDir, 16
 - State, 16
 - TotalJobs, 16
 - UpdateCredentials, 16
- ARex::ARexService, 16
- ARex::CountedResource, 22
- ARex::FileChunks, 28
 - Release, 28
 - Remove, 28
- ARex::FileChunksList, 28
 - Get, 29
 - GetStuck, 29
- ARex::GridManager, 32
- ARex::JobRecord, 38
- ARex::LoggerClient, 41
- ARex::OptimizedInformationContainer, 42
- ARex::PayloadBigFile, 43
 - ~PayloadBigFile, 44
 - PayloadBigFile, 44
- ARex::PayloadFile, 44
 - ~PayloadFile, 44
 - PayloadFile, 44
- ARexJob
 - ARex::ARexJob, 14
- AuthEvaluator, 17
- AuthUser, 17
- AuthVO, 17
- Cache::CacheService, 19
 - ~CacheService, 19
 - CacheCheck, 20
 - CacheLink, 20
 - CacheService, 19

- operator bool, 20
- process, 20
- RegistrationCollector, 20
- CacheCheck
 - Cache::CacheService, 20
- CacheConfig, 17
 - CacheConfig, 18
 - parseINIConf, 18
 - setCacheDirs, 18
- CacheConfigException, 18
- CacheLink
 - Cache::CacheService, 20
- CacheService
 - Cache::CacheService, 19
- Cancel
 - ARex::ARexJob, 14
- cancelJob
 - DTRGenerator, 26
- cap
 - voms_attrs, 55
- checkUploadedFiles
 - DTRGenerator, 26
- ChooseSessionDir
 - ARex::ARexJob, 14
- Clean
 - ARex::ARexJob, 14
- CommFIFO, 21
- ConfigSections, 21
- ContinuationPlugins, 22
- ContinuationPlugins::result_t, 47
- CreateFile
 - ARex::ARexJob, 15
- deploy
 - Janitor, 36
- DirectAccess, 22
- DirectAccess::diraccess_t, 22
- DirectFilePlugin, 23
- DirEntry, 23
- DREService, 11
- DREService::DREWebService, 23
 - ~DREWebService, 24
 - DREWebService, 24
 - logger, 24
 - makeFault, 24
 - ns_, 24
 - process, 24
- DREService::PerlProcessor, 45
 - ~PerlProcessor, 46
 - PerlProcessor, 46
- DREService::Task, 51
 - ~Task, 51
 - Task, 51
- DREService::TaskQueue, 51
 - ~TaskQueue, 52
 - pushTask, 52
 - shiftTask, 52
 - TaskQueue, 52
- DREService::TaskSet, 52
 - ~TaskSet, 52
 - removeTask, 53
 - TaskSet, 52
- DREWebService
 - DREService::DREWebService, 24
- DTRGenerator, 25
 - ~DTRGenerator, 25
 - cancelJob, 26
 - checkUploadedFiles, 26
 - DTRGenerator, 25
 - queryJobFinished, 26
 - receiveDTR, 26
 - receiveJob, 26
- DTRInfo, 27
 - DTRInfo, 27
- Entry, 27
- Failed
 - ARex::ARexJob, 15
- Failure
 - ARex::ARexJob, 15
- FileData, 29
- FileNode, 29
- FilePlugin, 30
- FileRoot, 30
- FileRoot::ServerParams, 48
- GACLPlugin, 30
- Get
 - ARex::FileChunksList, 29
- GetDescription
 - ARex::ARexJob, 15
- GetStuck
 - ARex::FileChunksList, 29
- gm_dirs_, 31
- GMEEnvironment, 31
 - nordugrid_config_loc, 31
 - support_mail_address, 31
- GridFTP_Commands, 32
- GridFTP_Commands_timeout, 32

- gridftp::ConfigSections, 21
- gridftp::Daemon, 22
- gridftp::GMEEnvironment, 31
 - nordugrid_config_loc, 31
 - support_mail_address, 31
- gridftp::LdapQuery, 39
 - ~LdapQuery, 40
 - Host, 40
 - LdapQuery, 40
 - Query, 40
 - Result, 40
 - Scope, 39
- gridftp::LdapQueryError, 40
 - LdapQueryError, 41
- gridftp::ParallelLdapQueries, 43
- gridftp::RunPlugin, 47
- gridftp::RunPlugin::lib_plugin_t, 41
- gridftp::UrlMapConfig, 53
- group
 - voms_attrs, 55
- Hopi::Hopi, 32
- Hopi::PayloadBigFile, 43
 - ~PayloadBigFile, 43
 - PayloadBigFile, 43
- Hopi::PayloadFile, 45
 - ~PayloadFile, 45
 - PayloadFile, 45
- Host
 - gridftp::LdapQuery, 40
- ID
 - ARex::ARexJob, 15
- Identity, 33
- Identity::Item, 35
- IdentityGACL, 33
- IdentityItemDN, 33
- IdentityItemVOMS, 34
- Index, 34
- ISIS::ISISService, 34
- ISIS::ISISSecAttr, 34
- ISIS::Neighbor_Container, 41
- Janitor, 35
 - deploy, 36
 - Janitor, 36
 - remove, 36
 - result, 36
 - wait, 36
- job_state_rec_t, 37
- JobDescription, 37
- JobLocalDescription, 37
- JobLog, 37
- JobPlugin, 37
- Jobs
 - ARex::ARexJob, 15
- JobsList, 38
- JobsListConfig, 38
- JobUser, 38
- JobUserHelper, 39
- JobUsers, 39
- LdapQuery
 - gridftp::LdapQuery, 40
- LdapQueryError
 - gridftp::LdapQueryError, 41
- LogDir
 - ARex::ARexJob, 15
- LogFiles
 - ARex::ARexJob, 15
- logger
 - DREService::DREWebService, 24
- LRMSResult, 41
- makeFault
 - DREService::DREWebService, 24
- nordugrid_config_loc
 - GMEEnvironment, 31
 - gridftp::GMEEnvironment, 31
- ns_
 - DREService::DREWebService, 24
- numvalue_for_shell, 42
- ObjectAccess, 42
- ObjectAccess::Item, 34
- ObjectAccessGACL, 42
- OpenDir
 - ARex::ARexJob, 15
- OpenFile
 - ARex::ARexJob, 15
- OpenLogFile
 - ARex::ARexJob, 16
- operator bool
 - Cache::CacheService, 20
- parseINICnf
 - CacheConfig, 18
- PayloadBigFile
 - ARex::PayloadBigFile, 44
 - Hopi::PayloadBigFile, 43

- PayloadFile
 - ARex::PayloadFile, 44
 - Hopi::PayloadFile, 45
- PerlProcessor
 - DREService::PerlProcessor, 46
- Permission, 46
- PermissionGACL, 46
- Policy, 47
- process
 - Arc::Service_JavaWrapper, 49
 - Arc::Service_PythonWrapper, 49
 - Cache::CacheService, 20
 - DREService::DREWebService, 24
 - SPService::Service_SP, 51
- pushTask
 - DREService::TaskQueue, 52
- Query
 - gridftpd::LdapQuery, 40
- queryJobFinished
 - DTRGenerator, 26
- receiveDTR
 - DTRGenerator, 26
- receiveJob
 - DTRGenerator, 26
- RegistrationCollector
 - Cache::CacheService, 20
- Release
 - ARex::FileChunks, 28
- Remove
 - ARex::FileChunks, 28
- remove
 - Janitor, 36
- removeTask
 - DREService::TaskSet, 53
- Result
 - gridftpd::LdapQuery, 40
- result
 - Janitor, 36
- Resume
 - ARex::ARexJob, 16
- role
 - voms_attrs, 55
- RunFunction, 47
- RunParallel, 47
- RunPlugin, 48
- RunPlugin::lib_plugin_t, 41
- RunPlugins, 48
- RunRedirected, 48
- Scope
 - gridftpd::LdapQuery, 39
- Server, 48
- server
 - voms, 54
- Service_SP
 - SPService::Service_SP, 50
- SessionDir
 - ARex::ARexJob, 16
- setCacheDirs
 - CacheConfig, 18
- shiftTask
 - DREService::TaskQueue, 52
- SimpleMap, 51
- SPService::Service_SP, 50
 - process, 51
 - Service_SP, 50
- State
 - ARex::ARexJob, 16
- std
 - voms, 54
- support_mail_address
 - GMEnvironment, 31
 - gridftpd::GMEnvironment, 31
- Task
 - DREService::Task, 51
- TaskQueue
 - DREService::TaskQueue, 52
- TaskSet
 - DREService::TaskSet, 52
- TotalJobs
 - ARex::ARexJob, 16
- UnixMap, 53
- UpdateCredentials
 - ARex::ARexJob, 16
- UrlMapConfig, 53
- userspec_t, 53
- value_for_shell, 54
- voms, 54
 - server, 54
 - std, 54
 - voname, 54
- voms_attrs, 54
 - cap, 55
 - group, 55
 - role, 55
- voname

voms, 54

wait

Janitor, 36

ZeroUInt, 55