

Hosting Environment (Daemon) Services

Generated by Doxygen 1.7.3

Mon Mar 7 2011 05:13:46

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	File Index	11
4.1	File List	11
5	Namespace Documentation	15
5.1	DREService Namespace Reference	15
5.1.1	Detailed Description	15
6	Data Structure Documentation	17
6.1	ARex::ARexGMConfig Class Reference	17
6.2	ARex::ARexJob Class Reference	17
6.2.1	Detailed Description	18
6.2.2	Constructor & Destructor Documentation	18
6.2.2.1	ARexJob	18
6.2.2.2	ARexJob	18
6.2.3	Member Function Documentation	18
6.2.3.1	Cancel	18
6.2.3.2	ChooseSessionDir	18
6.2.3.3	Clean	19
6.2.3.4	CreateFile	19
6.2.3.5	Failed	19
6.2.3.6	Failure	19
6.2.3.7	GetDescription	19
6.2.3.8	ID	19
6.2.3.9	Jobs	19
6.2.3.10	LogDir	19
6.2.3.11	LogFiles	19
6.2.3.12	OpenDir	19
6.2.3.13	OpenFile	20
6.2.3.14	OpenLogFile	20
6.2.3.15	Resume	20

6.2.3.16	SessionDir	20
6.2.3.17	State	20
6.2.3.18	State	20
6.2.3.19	TotalJobs	20
6.2.3.20	UpdateCredentials	20
6.3	ARex::ARexService Class Reference	20
6.4	AuthEvaluator Class Reference	21
6.5	AuthUser Class Reference	21
6.6	AuthVO Class Reference	21
6.7	CacheConfig Class Reference	21
6.7.1	Detailed Description	22
6.7.2	Constructor & Destructor Documentation	22
6.7.2.1	CacheConfig	22
6.7.2.2	CacheConfig	22
6.7.3	Member Function Documentation	22
6.7.3.1	parseINIConf	22
6.7.3.2	setCacheDirs	22
6.8	CacheConfigException Class Reference	22
6.8.1	Detailed Description	22
6.9	Cache::CacheService Class Reference	23
6.9.1	Detailed Description	23
6.9.2	Constructor & Destructor Documentation	23
6.9.2.1	CacheService	23
6.9.2.2	~CacheService	23
6.9.3	Member Function Documentation	24
6.9.3.1	CacheCheck	24
6.9.3.2	CacheLink	24
6.9.3.3	operator bool	24
6.9.3.4	operator!	24
6.9.3.5	process	24
6.9.3.6	RegistrationCollector	24
6.10	ArcSec::Charon Class Reference	25
6.10.1	Detailed Description	25
6.11	CommFIFO Class Reference	25
6.12	ARex::ConfGrp Class Reference	25
6.13	ARex::Config Class Reference	25
6.13.1	Detailed Description	26
6.13.2	Member Function Documentation	26
6.13.2.1	ConfValue	26
6.13.2.2	FirstConfValue	26
6.13.2.3	GetConfigs	26
6.14	ARex::ConfigError Class Reference	26
6.14.1	Detailed Description	26
6.14.2	Constructor & Destructor Documentation	26
6.14.2.1	ConfigError	26
6.15	ARex::ConfigIO Class Reference	27
6.15.1	Detailed Description	27
6.15.2	Member Function Documentation	27
6.15.2.1	Read	27
6.15.2.2	Write	27

6.16	ConfigSections Class Reference	27
6.17	gridftpd::ConfigSections Class Reference	28
6.18	ContinuationPlugins Class Reference	28
6.19	ARex::CountedResource Class Reference	28
6.20	gridftpd::Daemon Class Reference	28
6.21	DirectAccess::diraccess_t Struct Reference	28
6.22	DirectAccess Class Reference	28
6.23	DirectFilePlugin Class Reference	29
6.24	DirEntry Class Reference	29
6.25	DREService::DREWebService Class Reference	29
6.25.1	Constructor & Destructor Documentation	30
6.25.1.1	DREWebService	30
6.25.1.2	~DREWebService	30
6.25.2	Member Function Documentation	30
6.25.2.1	makeFault	30
6.25.2.2	process	30
6.25.3	Field Documentation	31
6.25.3.1	logger	31
6.25.3.2	ns_	31
6.26	DTRGenerator Class Reference	31
6.26.1	Detailed Description	31
6.26.2	Constructor & Destructor Documentation	31
6.26.2.1	DTRGenerator	31
6.26.2.2	~DTRGenerator	32
6.26.3	Member Function Documentation	32
6.26.3.1	cancelJob	32
6.26.3.2	checkUploadedFiles	32
6.26.3.3	queryJobFinished	32
6.26.3.4	receiveDTR	33
6.26.3.5	receiveJob	33
6.27	DTRInfo Class Reference	33
6.27.1	Detailed Description	33
6.27.2	Constructor & Destructor Documentation	33
6.27.2.1	DTRInfo	33
6.28	Entry Class Reference	34
6.29	ARex::FileChunks Class Reference	34
6.29.1	Detailed Description	34
6.29.2	Member Function Documentation	34
6.29.2.1	Release	34
6.29.2.2	Remove	34
6.30	ARex::FileChunksList Class Reference	35
6.30.1	Detailed Description	35
6.30.2	Member Function Documentation	35
6.30.2.1	Get	35
6.30.2.2	GetStuck	35
6.31	FileData Class Reference	35
6.32	FileNode Class Reference	35
6.33	FilePlugin Class Reference	36
6.34	FileRoot Class Reference	36
6.35	GACLPlugin Class Reference	36

6.36	gm_dirs_ Struct Reference	37
6.37	GMEEnvironment Class Reference	37
6.37.1	Member Function Documentation	37
6.37.1.1	nordugrid_config_loc	37
6.37.1.2	support_mail_address	37
6.38	gridftpd::GMEEnvironment Class Reference	37
6.38.1	Member Function Documentation	37
6.38.1.1	nordugrid_config_loc	37
6.38.1.2	support_mail_address	38
6.39	GridFTP_Commands Class Reference	38
6.40	GridFTP_Commands_timeout Class Reference	38
6.41	ARex::GridManager Class Reference	38
6.42	Hopi::Hopi Class Reference	38
6.43	Identity Class Reference	39
6.44	IdentityGACL Class Reference	39
6.45	IdentityItemDN Class Reference	39
6.46	IdentityItemVOMS Class Reference	40
6.47	Index Class Reference	40
6.48	ISIS::ISIService Class Reference	40
6.49	ISIS::ISISecAttr Class Reference	40
6.50	Identity::Item Class Reference	40
6.51	ObjectAccess::Item Class Reference	41
6.52	Janitor Class Reference	41
6.52.1	Detailed Description	42
6.52.2	Constructor & Destructor Documentation	42
6.52.2.1	Janitor	42
6.52.3	Member Function Documentation	42
6.52.3.1	deploy	42
6.52.3.2	remove	42
6.52.3.3	result	42
6.52.3.4	wait	42
6.53	job_state_rec_t Struct Reference	43
6.54	JobDescription Class Reference	43
6.55	JobLocalDescription Class Reference	43
6.56	JobLog Class Reference	43
6.56.1	Detailed Description	43
6.57	JobPlugin Class Reference	43
6.58	ARex::JobRecord Class Reference	44
6.59	JobsList Class Reference	44
6.60	JobsListConfig Class Reference	44
6.60.1	Detailed Description	44
6.61	JobUser Class Reference	44
6.62	JobUserHelper Class Reference	45
6.63	JobUsers Class Reference	45
6.64	gridftpd::LdapQuery Class Reference	45
6.64.1	Detailed Description	45
6.64.2	Member Enumeration Documentation	45
6.64.2.1	Scope	45
6.64.3	Constructor & Destructor Documentation	46
6.64.3.1	LdapQuery	46

6.64.3.2	~LdapQuery	46
6.64.4	Member Function Documentation	46
6.64.4.1	Host	46
6.64.4.2	Query	46
6.64.4.3	Result	46
6.65	gridftp::LdapQueryError Class Reference	46
6.65.1	Detailed Description	47
6.65.2	Constructor & Destructor Documentation	47
6.65.2.1	LdapQueryError	47
6.66	RunPlugin::lib_plugin_t Union Reference	47
6.67	gridftp::RunPlugin::lib_plugin_t Union Reference	47
6.68	ARex::LoggerClient Class Reference	47
6.69	LRMSResult Class Reference	47
6.70	ISIS::Neighbor_Container Class Reference	47
6.71	ARex::NGConfig Class Reference	48
6.71.1	Detailed Description	48
6.71.2	Member Function Documentation	48
6.71.2.1	Read	48
6.71.2.2	Write	48
6.72	numvalue_for_shell Class Reference	48
6.73	ObjectAccess Class Reference	49
6.74	ObjectAccessGACL Class Reference	49
6.75	ARex::OptimizedInformationContainer Class Reference	49
6.76	ARex::Option Class Reference	49
6.77	gridftp::ParallelLdapQueries Class Reference	50
6.77.1	Detailed Description	50
6.78	Hopi::PayloadBigFile Class Reference	50
6.78.1	Constructor & Destructor Documentation	50
6.78.1.1	PayloadBigFile	50
6.78.1.2	~PayloadBigFile	50
6.79	ARex::PayloadBigFile Class Reference	50
6.79.1	Constructor & Destructor Documentation	51
6.79.1.1	PayloadBigFile	51
6.79.1.2	~PayloadBigFile	51
6.80	ARex::PayloadFile Class Reference	51
6.80.1	Detailed Description	51
6.80.2	Constructor & Destructor Documentation	51
6.80.2.1	PayloadFile	51
6.80.2.2	~PayloadFile	51
6.81	Hopi::PayloadFile Class Reference	52
6.81.1	Detailed Description	52
6.81.2	Constructor & Destructor Documentation	52
6.81.2.1	PayloadFile	52
6.81.2.2	~PayloadFile	52
6.82	DREService::PerlProcessor Class Reference	52
6.82.1	Constructor & Destructor Documentation	53
6.82.1.1	PerlProcessor	53
6.82.1.2	~PerlProcessor	53
6.83	Permission Class Reference	53
6.84	PermissionGACL Class Reference	53

6.85 Policy Class Reference	54
6.86 ArcSec::Charon::PolicyLocation Class Reference	54
6.87 ContinuationPlugins::result_t Class Reference	54
6.88 RunFunction Class Reference	54
6.89 RunParallel Class Reference	54
6.90 RunPlugin Class Reference	54
6.91 gridftpd::RunPlugin Class Reference	55
6.92 RunPlugins Class Reference	55
6.93 RunRedirected Class Reference	55
6.94 Server Class Reference	55
6.95 FileRoot::ServerParams Class Reference	55
6.96 ArcSec::Service_AA Class Reference	55
6.96.1 Detailed Description	56
6.97 Arc::Service_JavaWrapper Class Reference	56
6.97.1 Member Function Documentation	56
6.97.1.1 process	56
6.98 Arc::Service_PythonWrapper Class Reference	56
6.98.1 Member Function Documentation	56
6.98.1.1 process	56
6.99 ArcSec::Service_SLCS Class Reference	57
6.99.1 Detailed Description	57
6.100SPService::Service_SP Class Reference	57
6.100.1 Detailed Description	57
6.100.2 Constructor & Destructor Documentation	57
6.100.2.1 Service_SP	57
6.100.3 Member Function Documentation	58
6.100.3.1 process	58
6.101SimpleMap Class Reference	58
6.102DREService::Task Class Reference	58
6.102.1 Constructor & Destructor Documentation	58
6.102.1.1 Task	58
6.102.1.2 ~Task	58
6.103DREService::TaskQueue Class Reference	58
6.103.1 Constructor & Destructor Documentation	59
6.103.1.1 TaskQueue	59
6.103.1.2 ~TaskQueue	59
6.103.2 Member Function Documentation	59
6.103.2.1 pushTask	59
6.103.2.2 shiftTask	59
6.104DREService::TaskSet Class Reference	59
6.104.1 Constructor & Destructor Documentation	59
6.104.1.1 TaskSet	59
6.104.1.2 ~TaskSet	60
6.104.2 Member Function Documentation	60
6.104.2.1 removeTask	60
6.105UnixMap Class Reference	60
6.106gridftpd::UrlMapConfig Class Reference	60
6.107UrlMapConfig Class Reference	60
6.108userspec_t Class Reference	60
6.109value_for_shell Class Reference	61

6.110voms Struct Reference	61
6.110.1 Detailed Description	61
6.110.2 Field Documentation	61
6.110.2.1 server	61
6.110.2.2 std	61
6.110.2.3 voname	61
6.111voms_attrs Struct Reference	61
6.111.1 Detailed Description	62
6.111.2 Field Documentation	62
6.111.2.1 cap	62
6.111.2.2 group	62
6.111.2.3 role	62
6.112ARex::XMLConfig Class Reference	62
6.112.1 Detailed Description	63
6.112.2 Member Function Documentation	63
6.112.2.1 Read	63
6.112.2.2 Write	63
6.113ZeroUInt Class Reference	63
6.113.1 Detailed Description	63
7 File Documentation	65
7.1 configcore.h File Reference	65
7.1.1 Detailed Description	65

Chapter 1

Namespace Index

1.1 Namespace List

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

DREService	15
-----------------------------	----

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

ARex::ARexGMConfig	17
ARex::ARexJob	17
ARex::ARexService	20
AuthEvaluator	21
AuthUser	21
AuthVO	21
CacheConfig	21
CacheConfigException	22
Cache::CacheService	23
ArcSec::Charon	25
CommFIFO	25
ARex::ConfGrp	25
ARex::Config	25
ARex::ConfigError	26
ARex::ConfigIO	27
ARex::NGConfig	48
ARex::XMLConfig	62
ConfigSections	27
gridftpd::ConfigSections	28
ContinuationPlugins	28
ARex::CountedResource	28
gridftpd::Daemon	28
DirectAccess::diraccess_t	28
DirectAccess	28
DirEntry	29
DREService::DREWebService	29
DTRGenerator	31
DTRInfo	33
Entry	34

ARex::FileChunks	34
ARex::FileChunksList	35
FileData	35
FileNode	35
FilePlugin	36
DirectFilePlugin	29
GACLPlugin	36
JobPlugin	43
FileRoot	36
gm_dirs_	37
GMEEnvironment	37
gridftpd::GMEEnvironment	37
GridFTP_Commands	38
GridFTP_Commands_timeout	38
ARex::GridManager	38
Hopi::Hopi	38
Identity	39
IdentityGACL	39
Index	40
ISIS::ISIService	40
ISIS::ISISecAttr	40
Identity::Item	40
IdentityItemDN	39
IdentityItemVOMS	40
ObjectAccess::Item	41
Janitor	41
job_state_rec_t	43
JobDescription	43
JobLocalDescription	43
JobLog	43
ARex::JobRecord	44
JobsList	44
JobsListConfig	44
JobUser	44
JobUserHelper	45
JobUsers	45
gridftpd::LdapQuery	45
gridftpd::LdapQueryError	46
RunPlugin::lib_plugin_t	47
gridftpd::RunPlugin::lib_plugin_t	47
ARex::LoggerClient	47
LRMSResult	47
ISIS::Neighbor_Container	47
numvalue_for_shell	48
ObjectAccess	49
ObjectAccessGACL	49
ARex::OptimizedInformationContainer	49
ARex::Option	49

gridftpd::ParallelLdapQueries	50
Hopi::PayloadBigFile	50
ARex::PayloadBigFile	50
ARex::PayloadFile	51
Hopi::PayloadFile	52
DREService::PerlProcessor	52
Permission	53
PermissionGACL	53
Policy	54
ArcSec::Charon::PolicyLocation	54
ContinuationPlugins::result_t	54
RunFunction	54
RunParallel	54
RunPlugin	54
gridftpd::RunPlugin	55
RunPlugins	55
RunRedirected	55
Server	55
FileRoot::ServerParams	55
ArcSec::Service_AA	55
Arc::Service_JavaWrapper	56
Arc::Service_PythonWrapper	56
ArcSec::Service_SLCS	57
SPService::Service_SP	57
SimpleMap	58
DREService::Task	58
DREService::TaskQueue	58
DREService::TaskSet	59
UnixMap	60
gridftpd::UrlMapConfig	60
UrlMapConfig	60
userspec_t	60
value_for_shell	61
voms	61
voms_attrs	61
ZeroUInt	63

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

ARex::ARexGMConfig	17
ARex::ARexJob	17
ARex::ARexService	20
AuthEvaluator	21
AuthUser	21
AuthVO	21
CacheConfig	21
CacheConfigException	22
Cache::CacheService	23
ArcSec::Charon	25
CommFIFO	25
ARex::ConfGrp	25
ARex::Config	25
ARex::ConfigError	26
ARex::ConfigIO	27
ConfigSections	27
gridftpd::ConfigSections	28
ContinuationPlugins	28
ARex::CountedResource	28
gridftpd::Daemon	28
DirectAccess::diraccess_t	28
DirectAccess	28
DirectFilePlugin	29
DirEntry	29
DREService::DREWebService	29
DTRGenerator	31
DTRInfo	33
Entry	34
ARex::FileChunks (Representation of delivered file chunks)	34

ARex::FileChunksList (Container for FileChunks (p. 34) instances)	35
FileData	35
FileNode	35
FilePlugin	36
FileRoot	36
GACLPlugin	36
gm_dirs	37
GMEEnvironment	37
gridftpd::GMEEnvironment	37
GridFTP_Commands	38
GridFTP_Commands_timeout	38
ARex::GridManager	38
Hopi::Hopi	38
Identity	39
IdentityGACL	39
IdentityItemDN	39
IdentityItemVOMS	40
Index	40
ISIS::ISISService	40
ISIS::ISISSecAttr	40
Identity::Item	40
ObjectAccess::Item	41
Janitor (Class to communicate with Janitor (p. 41) - Dynmaic Runtime Environment handler)	41
job_state_rec_t	43
JobDescription	43
JobLocalDescription	43
JobLog	43
JobPlugin	43
ARex::JobRecord	44
JobsList	44
JobsListConfig	44
JobUser	44
JobUserHelper	45
JobUsers	45
gridftpd::LdapQuery	45
gridftpd::LdapQueryError	46
RunPlugin::lib_plugin_t	47
gridftpd::RunPlugin::lib_plugin_t	47
ARex::LoggerClient	47
LRMSResult	47
ISIS::Neighbor_Container	47
ARex::NGConfig	48
numvalue_for_shell	48
ObjectAccess	49
ObjectAccessGACL	49
ARex::OptimizedInformationContainer	49
ARex::Option	49
gridftpd::ParallelLdapQueries	50
Hopi::PayloadBigFile	50

ARex::PayloadBigFile	50
ARex::PayloadFile	51
Hopi::PayloadFile	52
DREService::PerlProcessor	52
Permission	53
PermissionGACL	53
Policy	54
ArcSec::Charon::PolicyLocation	54
ContinuationPlugins::result_t	54
RunFunction	54
RunParallel	54
RunPlugin	54
gridftpd::RunPlugin	55
RunPlugins	55
RunRedirected	55
Server	55
FileRoot::ServerParams	55
ArcSec::Service_AA	55
Arc::Service_JavaWrapper	56
Arc::Service_PythonWrapper	56
ArcSec::Service_SLCS	57
SPService::Service_SP	57
SimpleMap	58
DREService::Task	58
DREService::TaskQueue	58
DREService::TaskSet	59
UnixMap	60
gridftpd::UrlMapConfig	60
UrlMapConfig	60
userspec_t	60
value_for_shell	61
voms	61
voms_attrs	61
ARex::XMLConfig	62
ZeroUInt	63

Chapter 4

File Index

4.1 File List

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

aaservice.h	??
arex.h	??
auth.h	??
CacheService.h	??
a-rex/grid-manager/misc/canonical_dir.h	??
gridftpd/misc/canonical_dir.h	??
charon.h	??
client.h	??
commands.h	??
commfifo.h	??
a-rex/grid-manager/conf/conf.h	??
gridftpd/conf/conf.h	??
gridftpd/conf.h	??
conf_cache.h	??
conf_file.h	??
a-rex/grid-manager/conf/conf_map.h	??
gridftpd/conf/conf_map.h	??
conf_pre.h	??
a-rex/grid-manager/conf/conf_sections.h	??
gridftpd/conf/conf_sections.h	??
conf_vo.h	??
configcore.h	65
configio.h	??
daemon.h	??
delete.h	??
dREWebService.h	??
dtr_generator.h	??
Entry.h	??
a-rex/grid-manager/conf/environment.h	??

gridftpd/conf/environment.h	??
a-rex/grid-manager/misc/escaped.h	??
gridftpd/misc/escaped.h	??
FileChunks.h	??
fileplugin.h	??
fileroot.h	??
a-rex/gacl.h	??
gridftpd/external/gacl/gacl.h	??
gacl_auth.h	??
gaclplugin.h	??
grid_manager.h	??
a-rex/grid-manager/conf/gridmap.h	??
gridftpd/conf/gridmap.h	??
hopi.h	??
identity.h	??
identity_dn.h	??
identity_gacl.h	??
identity_voms.h	??
Index.h	??
info_files.h	??
info_log.h	??
info_types.h	??
isis.h	??
janitor.h	??
javawrapper.h	??
grid-manager/jobs/job.h	??
job.h	??
job_desc.h	??
job_log.h	??
job_request.h	??
jobplugin.h	??
JobRecord.h	??
ldapquery.h	??
LDIFtoXML.h	??
misc.h	??
names.h	??
ngconfig.h	??
object_access.h	??
object_access_gacl.h	??
a-rex/PayloadFile.h	??
hopi/PayloadFile.h	??
PerlProcessor.h	??
permission.h	??
permission_gacl.h	??
plugins.h	??
Policy.h	??
a-rex/grid-manager/misc/proxy.h	??
gridftpd/misc/proxy.h	??
pythonwrapper.h	??
run_function.h	??

run_parallel.h	??
a-rex/grid-manager/run/run_plugin.h	??
gridftpd/run/run_plugin.h	??
run_redirected.h	??
security.h	??
send_mail.h	??
Server.h	??
simplemap.h	??
slcs.h	??
SPService.h	??
states.h	??
Task.h	??
TaskQueue.h	??
TaskSet.h	??
tools.h	??
unixmap.h	??
users.h	??
userspec.h	??
util++.h	??
util.h	??
xmlconfig.h	??

Chapter 5

Namespace Documentation

5.1 DREService Namespace Reference

Data Structures

- class **DREWebService**
- class **PerlProcessor**
- class **Task**
- class **TaskQueue**
- class **TaskSet**

5.1.1 Detailed Description

Implementation of a simple echo service

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

Chapter 6

Data Structure Documentation

6.1 ARex::ARexGMConfig Class Reference

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

- job.h

6.2 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)

6.2.1 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.

6.2.2 Constructor & Destructor Documentation

- 6.2.2.1 **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

- 6.2.2.2 **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

6.2.3 Member Function Documentation

- 6.2.3.1 **bool ARex::ARexJob::Cancel** (void)

Cancel processing/execution of job

- 6.2.3.2 **bool ARex::ARexJob::ChooseSessionDir** (const std::string & *jobid*, std::string & *sessiondir*)

Select a session dir to use for this job

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

Remove job from local pool

6.2.3.4 int ARex::ARexJob::CreateFile (const std::string & filename)

Creates file in job's session directory and returns handler

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

Returns true if job has failed

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

Returns textual description of failure of last operation

6.2.3.7 bool ARex::ARexJob::GetDescription (Arc::XMLNode & jsdl)

Fills provided jsdl with job description

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

Return ID assigned to job

6.2.3.9 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.

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

Returns name of virtual log directory

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

Returns list of existing log files

6.2.3.12 Glib::Dir* ARex::ARexJob::OpenDir (const std::string & dirname)

Opens directory inside session directory

6.2.3.13 `int ARex::ARexJob::OpenFile (const std::string & filename, bool for_read, bool for_write)`

Opens file in job's session directory and returns handler

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

Opens log file in control directory

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

Resume execution of job after error

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

Returns path to session directory

6.2.3.17 `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

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

Returns current state of job

6.2.3.19 `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.

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

Updates job credentials

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

- `job.h`

6.3 ARex::ARexService Class Reference

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

- arex.h

6.4 AuthEvaluator Class Reference

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

- auth.h

6.5 AuthUser Class Reference

Data Structures

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

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

- auth.h

6.6 AuthVO Class Reference

Friends

- class **AuthUser**

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

- auth.h

6.7 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)

6.7.1 Detailed Description

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

6.7.2 Constructor & Destructor Documentation

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

Create a new **CacheConfig** (p. 21) 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.

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

Empty **CacheConfig** (p. 21)

6.7.3 Member Function Documentation

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

Parsers for the two different conf styles

6.7.3.2 `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`

6.8 CacheConfigException Class Reference

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

6.8.1 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`

6.9 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)

6.9.1 Detailed Description

CacheService (p. 23) 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.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 Cache::CacheService::CacheService (Arc::Config * cfg)

Make a new **CacheService** (p. 23). Reads the configuration and determines the validity of the service.

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

Destroy the **CacheService** (p. 23)

6.9.3 Member Function Documentation

6.9.3.1 **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
-------------	--

6.9.3.2 **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

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

Returns true if the **CacheService** (p. 23) is valid.

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

Returns true if the **CacheService** (p. 23) is not valid.

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

Main method called by HED when **CacheService** (p. 23) is invoked. Directs call to appropriate **CacheService** (p. 23) method.

6.9.3.6 **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

6.10 ArcSec::Charon Class Reference

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

Data Structures

- class **PolicyLocation**

6.10.1 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

6.11 CommFIFO Class Reference

Data Structures

- class **elem_t**

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

- commfifo.h

6.12 ARex::ConfGrp Class Reference

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

- configcore.h

6.13 ARex::Config Class Reference

```
#include <configcore.h>
```

Public Member Functions

- const std::list< **ConfGrp** > & **GetConfigs** () const
- std::list< std::string > **ConfValue** (const std::string &path) const
- std::string **FirstConfValue** (const std::string &path) const

6.13.1 Detailed Description

Core configuration class.

6.13.2 Member Function Documentation

6.13.2.1 `std::list<std::string> ARex::Config::ConfValue (const std::string & path) const`

Get the configuration values from key.

6.13.2.2 `std::string ARex::Config::FirstConfValue (const std::string & path) const`

Get the first configuration value from key. This is meant as a short cut when it is known that the key is not multivalued.

6.13.2.3 `const std::list<ConfGrp>& ARex::Config::GetConfigs () const`

Returns the parsed options.

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

- `configcore.h`

6.14 ARex::ConfigError Class Reference

```
#include <configcore.h>
```

Public Member Functions

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

6.14.1 Detailed Description

Error configuration class.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 `ARex::ConfigError::ConfigError (std::string message) [inline]`

Constructor for the **ConfigError** (p. 26) exception. Calls the corresponding constructor in ARCLibError.

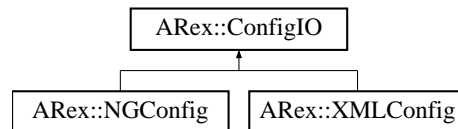
The documentation for this class was generated from the following file:

- `configcore.h`

6.15 ARex::ConfigIO Class Reference

```
#include <configio.h>
```

Inheritance diagram for ARex::ConfigIO:



Public Member Functions

- virtual **Config Read** (std::istream &is)=0
- virtual void **Write** (const **Config** &conf, std::ostream &os)=0

6.15.1 Detailed Description

Virtual base-class for reading and writing configuration files. Concrete instances include **NGConfig** (p. 48) and **XMLConfig** (p. 62).

6.15.2 Member Function Documentation

6.15.2.1 virtual Config ARex::ConfigIO::Read (std::istream & is) [pure virtual]

Read the named configuration source.

Implemented in **ARex::NGConfig** (p. 48), and **ARex::XMLConfig** (p. 63).

6.15.2.2 virtual void ARex::ConfigIO::Write (const Config & conf, std::ostream & os) [pure virtual]

Write configuration to named configuration destination.

Implemented in **ARex::NGConfig** (p. 48), and **ARex::XMLConfig** (p. 63).

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

- configio.h

6.16 ConfigSections Class Reference

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

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

6.17 gridftpd::ConfigSections Class Reference

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

- gridftpd/conf/conf_sections.h

6.18 ContinuationPlugins Class Reference

Data Structures

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

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

- plugins.h

6.19 ARex::CountedResource Class Reference

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

- arex.h

6.20 gridftpd::Daemon Class Reference

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

- daemon.h

6.21 DirectAccess::diraccess_t Struct Reference

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

- fileplugin.h

6.22 DirectAccess Class Reference

Data Structures

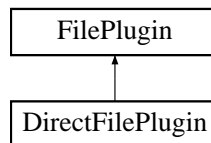
- struct **diraccess_t**

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

- fileplugin.h

6.23 DirectFilePlugin Class Reference

Inheritance diagram for DirectFilePlugin:



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

- fileplugin.h

6.24 DirEntry Class Reference

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

- fileroot.h

6.25 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**

6.25.1 Constructor & Destructor Documentation

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

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

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

Destructor.

6.25.2 Member Function Documentation

6.25.2.1 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

6.25.2.2 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>inmsg</i>	outgoing message

Returns

Status of the result achieved

6.25.3 Field Documentation

6.25.3.1 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.

6.25.3.2 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

6.26 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)
- int **checkUploadedFiles** (**JobDescription** &job)

6.26.1 Detailed Description

A-REX implementation of DTR Generator.

6.26.2 Constructor & Destructor Documentation

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

Start up Generator.

Parameters

<i>user</i>	JobUsers (p. 45) 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

6.26.2.2 DTRGenerator::~~DTRGenerator ()

Stop Generator

6.26.3 Member Function Documentation

6.26.3.1 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
------------	----------------------------------

6.26.3.2 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

6.26.3.3 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. 43) 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.

6.26.3.4 `virtual void DTRGenerator::receiveDTR (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
------------	---

6.26.3.5 `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`

6.27 DTRInfo Class Reference

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

Public Member Functions

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

6.27.1 Detailed Description

DTRInfo (p. 33) 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.

6.27.2 Constructor & Destructor Documentation**6.27.2.1 DTRInfo::DTRInfo (const JobUsers & *users*)**

JobUsers (p. 45) is needed to find the correct control dir

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

- `dtr_generator.h`

6.28 Entry Class Reference

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

- Entry.h

6.29 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, size_t csize)
- bool **Complete** (void)
- void **Print** (void)
- void **Release** (void)
- void **Remove** (void)

6.29.1 Detailed Description

Representation of delivered file chunks.

6.29.2 Member Function Documentation

6.29.2.1 void ARex::FileChunks::Release (void)

Release reference obtained through **FileChunksList::Get()** (p. 35) method. This operation may lead to destruction of FileChunk instance hence previously obtained reference must not be used.

6.29.2.2 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. 34) after whole file is delivered in order to free resources associated with **FileChunks** (p. 34) instance.

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

- FileChunks.h

6.30 ARex::FileChunksList Class Reference

Container for **FileChunks** (p. 34) instances.

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

Public Member Functions

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

6.30.1 Detailed Description

Container for **FileChunks** (p. 34) instances.

6.30.2 Member Function Documentation

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

Returns previously created **FileChunks** (p. 34) 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.

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

Returns pointer to first stuck file. File is considred 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

6.31 FileData Class Reference

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

- info_types.h

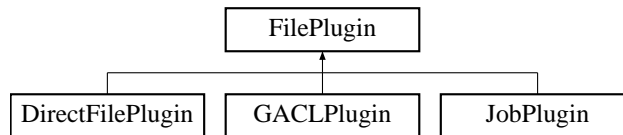
6.32 FileNode Class Reference

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

- fileroot.h

6.33 FilePlugin Class Reference

Inheritance diagram for FilePlugin:



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

- fileroot.h

6.34 FileRoot Class Reference

Data Structures

- class **ServerParams**

Friends

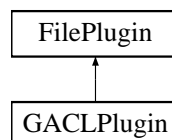
- class **GridFTP_Commands**

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

- fileroot.h

6.35 GACLPlugin Class Reference

Inheritance diagram for GACLPlugin:



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

- gacplugin.h

6.36 gm_dirs_ Struct Reference

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

- jobplugin.h

6.37 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

6.37.1 Member Function Documentation

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

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

6.37.1.2 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

6.38 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

6.38.1 Member Function Documentation

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

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

6.38.1.2 `std::string gridftp::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:

- `gridftp/conf/environment.h`

6.39 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`

6.40 GridFTP_Commands_timeout Class Reference

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

- `commands.h`

6.41 ARex::GridManager Class Reference

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

- `grid_manager.h`

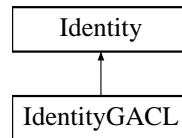
6.42 Hopi::Hopi Class Reference

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

- `hopi.h`

6.43 Identity Class Reference

Inheritance diagram for Identity:



Data Structures

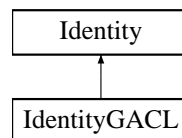
- class **Item**

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

- identity.h

6.44 IdentityGACL Class Reference

Inheritance diagram for IdentityGACL:

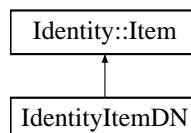


The documentation for this class was generated from the following file:

- identity_gacl.h

6.45 IdentityItemDN Class Reference

Inheritance diagram for IdentityItemDN:

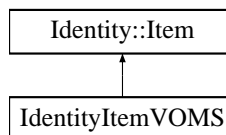


The documentation for this class was generated from the following file:

- identity_dn.h

6.46 IdentityItemVOMS Class Reference

Inheritance diagram for IdentityItemVOMS:



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

- identity_voms.h

6.47 Index Class Reference

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

- Index.h

6.48 ISIS::ISIService Class Reference

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

- isis.h

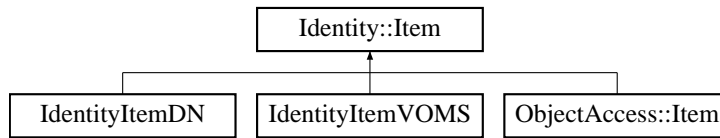
6.49 ISIS::ISISecAttr Class Reference

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

- security.h

6.50 Identity::Item Class Reference

Inheritance diagram for Identity::Item:

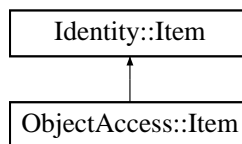


The documentation for this class was generated from the following file:

- identity.h

6.51 ObjectAccess::Item Class Reference

Inheritance diagram for ObjectAccess::Item:



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

- object_access.h

6.52 Janitor Class Reference

Class to communicate with **Janitor** (p. 41) - 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)

6.52.1 Detailed Description

Class to communicate with **Janitor** (p. 41) - Dynmaic Runtime Environment handler.

6.52.2 Constructor & Destructor Documentation

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

Creates instance representing job entry in **Janitor** (p. 41) database.

Takes id for job identifier and cdir for the control directory of A-Rex. constructor does not register job in the **Janitor** (p. 41). It only associates job with this instance.

6.52.3 Member Function Documentation

6.52.3.1 **bool Janitor::deploy** (`void`)

Registers associated job with **Janitor** (p. 41) and deploys dynamic RTEs.

This operation is asynchronous. Returned true means **Janitor** (p. 41) will be contacted and deployment will start soon. For obtaining result of operation see methods **wait()** (p. 42) and **result()** (p. 42). During this operation janitor utility is called with command register and optionally deploy.

6.52.3.2 **bool Janitor::remove** (`void`)

Removes job from those handled by **Janitor** (p. 41) and releases associated RTEs.

This operation is asynchronous. Returned true means **Janitor** (p. 41) will be contacted and removal will start soon. For obtaining result of operation see methods **wait()** (p. 42) and **result()** (p. 42). During this operation janitor utility is called with command remove.

6.52.3.3 **Result Janitor::result** (`void`)

Returns true if operation initiated by **deploy()** (p. 42) or **remove()** (p. 42) succeeded.

It should be called after **wait()** (p. 42) returned true.

6.52.3.4 **bool Janitor::wait** (`int timeout`)

Wait till operation initiated by **deploy()** (p. 42) or **remove()** (p. 42) 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`

6.53 `job_state_rec_t` Struct Reference

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

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

6.54 JobDescription Class Reference

Friends

- class `JobsList`

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

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

6.55 JobLocalDescription Class Reference

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

- `info_types.h`

6.56 JobLog Class Reference

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

6.56.1 Detailed Description

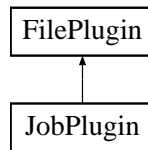
Put short information into log when every job starts/finishes. And store more detailed information for Reporter.

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

- `job_log.h`

6.57 JobPlugin Class Reference

Inheritance diagram for JobPlugin:



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

- jobplugin.h

6.58 ARex::JobRecord Class Reference

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

- JobRecord.h

6.59 JobsList Class Reference

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

- states.h

6.60 JobsListConfig Class Reference

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

Friends

- class **JobsList**

6.60.1 Detailed Description

Class to represent information read from configuration.

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

- states.h

6.61 JobUser Class Reference

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

- users.h

6.62 JobUserHelper Class Reference

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

- users.h

6.63 JobUsers Class Reference

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

- users.h

6.64 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** ()

6.64.1 Detailed Description

LdapQuery (p. 45) class; querying of LDAP servers.

6.64.2 Member Enumeration Documentation

6.64.2.1 enum gridftp::LdapQuery::Scope

Scope for a LDAP queries. Use when querying.

6.64.3 Constructor & Destructor Documentation

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

Constructs a new **LdapQuery** (p. 45) object and sets connection options. The connection is first established when calling `Query`.

6.64.3.2 `gridftp::LdapQuery::~~LdapQuery ()`

Destructor. Will disconnect from the ldapserver if still connected.

6.64.4 Member Function Documentation

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

Returns the hostname of the ldap-server.

6.64.4.2 `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.

6.64.4.3 `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`

6.65 gridftp::LdapQueryError Class Reference

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

Public Member Functions

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

6.65.1 Detailed Description

LdapQuery (p. 45) exception. Gets thrown when an error occurs in a query.

6.65.2 Constructor & Destructor Documentation

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

Standard exception class constructor.

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

- ldapquery.h

6.66 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

6.67 gridftpd::RunPlugin::lib_plugin_t Union Reference

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

- gridftpd/run/run_plugin.h

6.68 ARex::LoggerClient Class Reference

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

- client.h

6.69 LRMSResult Class Reference

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

- info_types.h

6.70 ISIS::Neighbor_Container Class Reference

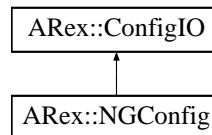
The documentation for this class was generated from the following file:

- isis.h

6.71 ARex::NGConfig Class Reference

```
#include <ngconfig.h>
```

Inheritance diagram for ARex::NGConfig:



Public Member Functions

- **Config Read** (std::istream &is)
- void **Write** (const **Config** &config, std::ostream &os)

6.71.1 Detailed Description

Configuration class used for reading configuration files ARC-style.

6.71.2 Member Function Documentation

6.71.2.1 Config ARex::NGConfig::Read (std::istream & is) [virtual]

Read old arc.conf style configuration.

Implements **ARex::ConfigIO** (p. 27).

6.71.2.2 void ARex::NGConfig::Write (const Config & config, std::ostream & os) [virtual]

Write configuration to named file.

Implements **ARex::ConfigIO** (p. 27).

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

- ngconfig.h

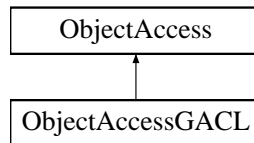
6.72 numvalue_for_shell Class Reference

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

- job_desc.h

6.73 ObjectAccess Class Reference

Inheritance diagram for ObjectAccess:



Data Structures

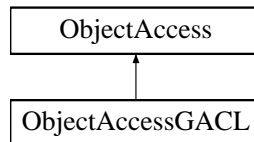
- class **Item**

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

- object_access.h

6.74 ObjectAccessGACL Class Reference

Inheritance diagram for ObjectAccessGACL:



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

- object_access_gacl.h

6.75 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

6.76 ARex::Option Class Reference

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

- configcore.h

6.77 gridftp::ParallelLdapQueries Class Reference

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

6.77.1 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

6.78 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

6.78.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.78.1.2 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

6.79 ARex::PayloadBigFile Class Reference

Public Member Functions

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

6.79.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.79.1.2 **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

6.80 ARex::PayloadFile Class Reference

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

Public Member Functions

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

6.80.1 Detailed Description

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

6.80.2 Constructor & Destructor Documentation

6.80.2.1 **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.

6.80.2.2 **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

6.81 Hopi::PayloadFile Class Reference

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

Public Member Functions

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

6.81.1 Detailed Description

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

6.81.2 Constructor & Destructor Documentation

6.81.2.1 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.

6.81.2.2 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

6.82 DREService::PerlProcessor Class Reference

Data Structures

- struct **ThreadInterface**

Public Member Functions

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

6.82.1 Constructor & Destructor Documentation

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

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

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

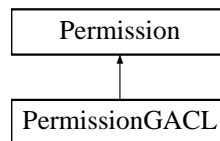
Destructor.

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

- PerlProcessor.h

6.83 Permission Class Reference

Inheritance diagram for Permission:

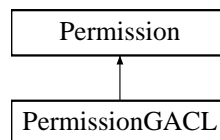


The documentation for this class was generated from the following file:

- permission.h

6.84 PermissionGACL Class Reference

Inheritance diagram for PermissionGACL:



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

- permission_gacl.h

6.85 Policy Class Reference

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

- Policy.h

6.86 ArcSec::Charon::PolicyLocation Class Reference

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

- charon.h

6.87 ContinuationPlugins::result_t Class Reference

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

- plugins.h

6.88 RunFunction Class Reference

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

- run_function.h

6.89 RunParallel Class Reference

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

- run_parallel.h

6.90 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

6.91 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`

6.92 RunPlugins Class Reference

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

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

6.93 RunRedirected Class Reference

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

- `run_redirected.h`

6.94 Server Class Reference

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

- `Server.h`

6.95 FileRoot::ServerParams Class Reference

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

- `fileroot.h`

6.96 ArcSec::Service_AA Class Reference

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

6.96.1 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

6.97 Arc::Service_JavaWrapper Class Reference

Public Member Functions

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

6.97.1 Member Function Documentation

6.97.1.1 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

6.98 Arc::Service_PythonWrapper Class Reference

Public Member Functions

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

6.98.1 Member Function Documentation

6.98.1.1 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

6.99 ArcSec::Service_SLCS Class Reference

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

6.99.1 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

6.100 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 &)

6.100.1 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

6.100.2 Constructor & Destructor Documentation

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

Constructor

6.100.3 Member Function Documentation

6.100.3.1 `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

6.101 SimpleMap Class Reference

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

- simplemap.h

6.102 DREService::Task Class Reference

Public Member Functions

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

6.102.1 Constructor & Destructor Documentation

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

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

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

Destructor.

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

- Task.h

6.103 DREService::TaskQueue Class Reference

Public Member Functions

- **TaskQueue** (int length)

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

6.103.1 Constructor & Destructor Documentation

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

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

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

Destructor.

6.103.2 Member Function Documentation

6.103.2.1 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.

6.103.2.2 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

6.104 DREService::TaskSet Class Reference

Public Member Functions

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

6.104.1 Constructor & Destructor Documentation

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

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

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

Destructor.

6.104.2 Member Function Documentation

6.104.2.1 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

6.105 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

6.106 gridftpd::UrlMapConfig Class Reference

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

- gridftpd/conf/conf_map.h

6.107 UrlMapConfig Class Reference

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

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

6.108 userspec_t Class Reference

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

- userspec.h

6.109 value_for_shell Class Reference

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

- job_desc.h

6.110 voms Struct Reference

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

Data Fields

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

6.110.1 Detailed Description

VOMS data

6.110.2 Field Documentation

6.110.2.1 std::string voms::server

The VOMS server DN, as from its certificate

6.110.2.2 std::vector<voms_attrs> voms::std

User's characteristics

6.110.2.3 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

6.111 voms_attrs Struct Reference

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

Data Fields

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

6.111.1 Detailed Description

VOMS attributes

6.111.2 Field Documentation

6.111.2.1 `std::string voms_attrs::cap`

user's capability

6.111.2.2 `std::string voms_attrs::group`

user's group

6.111.2.3 `std::string voms_attrs::role`

user's role

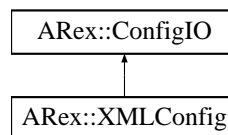
The documentation for this struct was generated from the following file:

- `auth.h`

6.112 ARex::XMLConfig Class Reference

```
#include <xmlconfig.h>
```

Inheritance diagram for ARex::XMLConfig:



Public Member Functions

- **Config Read** (`std::istream &is`)
- **void Write** (`const Config &config`, `std::ostream &os`)

6.112.1 Detailed Description

Class for reading in configuration files in xml-format. It uses libxml2 for xml-parsing.

6.112.2 Member Function Documentation

6.112.2.1 Config ARex::XMLConfig::Read (std::istream & *is*) [virtual]

Read configuration.

Implements **ARex::ConfigIO** (p. 27).

6.112.2.2 void ARex::XMLConfig::Write (const Config & *config*, std::ostream & *os*) [virtual]

Write configuration.

Implements **ARex::ConfigIO** (p. 27).

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

- xmlconfig.h

6.113 ZeroUInt Class Reference

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

6.113.1 Detailed Description

ZeroUInt (p. 63) 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

Chapter 7

File Documentation

7.1 configcore.h File Reference

```
#include <iostream>
#include <list>
#include <map>
#include <string>
#include <arc/Logger.h>
```

Data Structures

- class **ARex::ConfigError**
- class **ARex::Option**
- class **ARex::ConfGrp**
- class **ARex::Config**

Functions

- Config **ARex::ReadConfig** (std::istream &is)
- Config **ARex::ReadConfig** (const std::string &filename)

7.1.1 Detailed Description

This file describes the core configuration

Index

- ~CacheService
 - Cache::CacheService, 23
- ~DREWebService
 - DREService::DREWebService, 30
- ~DTRGenerator
 - DTRGenerator, 32
- ~LdapQuery
 - gridftpd::LdapQuery, 46
- ~PayloadBigFile
 - ARex::PayloadBigFile, 51
 - Hopi::PayloadBigFile, 50
- ~PayloadFile
 - ARex::PayloadFile, 51
 - Hopi::PayloadFile, 52
- ~PerlProcessor
 - DREService::PerlProcessor, 53
- ~Task
 - DREService::Task, 58
- ~TaskQueue
 - DREService::TaskQueue, 59
- ~TaskSet
 - DREService::TaskSet, 59
- Arc::Service_JavaWrapper, 56
 - process, 56
- Arc::Service_PythonWrapper, 56
 - process, 56
- ArcSec::Charon, 25
- ArcSec::Charon::PolicyLocation, 54
- ArcSec::Service_AA, 55
- ArcSec::Service_SLCS, 57
- ARex::ARexGMConfig, 17
- ARex::ARexJob, 17
 - ARexJob, 18
 - Cancel, 18
 - ChooseSessionDir, 18
 - Clean, 18
 - CreateFile, 19
 - Failed, 19
 - Failure, 19
 - GetDescription, 19
 - ID, 19
 - Jobs, 19
 - LogDir, 19
 - LogFiles, 19
 - OpenDir, 19
 - OpenFile, 19
 - OpenLogFile, 20
 - Resume, 20
 - SessionDir, 20
 - State, 20
 - TotalJobs, 20
 - UpdateCredentials, 20
- ARex::ARexService, 20
- ARex::ConfGrp, 25
- ARex::Config, 25
 - ConfValue, 26
 - FirstConfValue, 26
 - GetConfigs, 26
- ARex::ConfigError, 26
 - ConfigError, 26
- ARex::ConfigIO, 27
 - Read, 27
 - Write, 27
- ARex::CountedResource, 28
- ARex::FileChunks, 34
 - Release, 34
 - Remove, 34
- ARex::FileChunksList, 35
 - Get, 35
 - GetStuck, 35
- ARex::GridManager, 38
- ARex::JobRecord, 44
- ARex::LoggerClient, 47
- ARex::NGConfig, 48
 - Read, 48
 - Write, 48
- ARex::OptimizedInformationContainer, 49
- ARex::Option, 49
- ARex::PayloadBigFile, 50
 - ~PayloadBigFile, 51
 - PayloadBigFile, 51

- ARex::PayloadFile, 51
 - ~PayloadFile, 51
 - PayloadFile, 51
- ARex::XMLConfig, 62
 - Read, 63
 - Write, 63
- ARexJob
 - ARex::ARexJob, 18
- AuthEvaluator, 21
- AuthUser, 21
- AuthVO, 21
- Cache::CacheService, 23
 - ~CacheService, 23
 - CacheCheck, 24
 - CacheLink, 24
 - CacheService, 23
 - operator bool, 24
 - process, 24
 - RegistrationCollector, 24
- CacheCheck
 - Cache::CacheService, 24
- CacheConfig, 21
 - CacheConfig, 22
 - parseINIconf, 22
 - setCacheDirs, 22
- CacheConfigException, 22
- CacheLink
 - Cache::CacheService, 24
- CacheService
 - Cache::CacheService, 23
- Cancel
 - ARex::ARexJob, 18
- cancelJob
 - DTRGenerator, 32
- cap
 - voms_attrs, 62
- checkUploadedFiles
 - DTRGenerator, 32
- ChooseSessionDir
 - ARex::ARexJob, 18
- Clean
 - ARex::ARexJob, 18
- CommFIFO, 25
- configcore.h, 65
- ConfigError
 - ARex::ConfigError, 26
- ConfigSections, 27
- ConfValue
 - ARex::Config, 26
- ContinuationPlugins, 28
- ContinuationPlugins::result_t, 54
- CreateFile
 - ARex::ARexJob, 19
- deploy
 - Janitor, 42
- DirectAccess, 28
- DirectAccess::diraccess_t, 28
- DirectFilePlugin, 29
- DirEntry, 29
- DREService, 15
- DREService::DREWebService, 29
 - ~DREWebService, 30
 - DREWebService, 30
 - logger, 31
 - makeFault, 30
 - ns_, 31
 - process, 30
- DREService::PerlProcessor, 52
 - ~PerlProcessor, 53
 - PerlProcessor, 53
- DREService::Task, 58
 - ~Task, 58
 - Task, 58
- DREService::TaskQueue, 58
 - ~TaskQueue, 59
 - pushTask, 59
 - shiftTask, 59
 - TaskQueue, 59
- DREService::TaskSet, 59
 - ~TaskSet, 59
 - removeTask, 60
 - TaskSet, 59
- DREWebService
 - DREService::DREWebService, 30
- DTRGenerator, 31
 - ~DTRGenerator, 32
 - cancelJob, 32
 - checkUploadedFiles, 32
 - DTRGenerator, 31
 - queryJobFinished, 32
 - receiveDTR, 32
 - receiveJob, 33
- DTRInfo, 33
 - DTRInfo, 33
- Entry, 34
- Failed

- ARex::ARexJob, 19
- Failure
 - ARex::ARexJob, 19
- FileData, 35
- FileNode, 35
- FilePlugin, 36
- FileRoot, 36
- FileRoot::ServerParams, 55
- FirstConfValue
 - ARex::Config, 26
- GACLPlugin, 36
- Get
 - ARex::FileChunksList, 35
- GetConfigs
 - ARex::Config, 26
- GetDescription
 - ARex::ARexJob, 19
- GetStuck
 - ARex::FileChunksList, 35
- gm_dirs_, 37
- GMEEnvironment, 37
 - nordugrid_config_loc, 37
 - support_mail_address, 37
- GridFTP_Commands, 38
- GridFTP_Commands_timeout, 38
- gridftpd::ConfigSections, 28
- gridftpd::Daemon, 28
- gridftpd::GMEEnvironment, 37
 - nordugrid_config_loc, 37
 - support_mail_address, 37
- gridftpd::LdapQuery, 45
 - ~LdapQuery, 46
 - Host, 46
 - LdapQuery, 46
 - Query, 46
 - Result, 46
 - Scope, 45
- gridftpd::LdapQueryError, 46
 - LdapQueryError, 47
- gridftpd::ParallelLdapQueries, 50
- gridftpd::RunPlugin, 55
- gridftpd::RunPlugin::lib_plugin_t, 47
- gridftpd::UrlMapConfig, 60
- group
 - voms_attrs, 62
- Hopi::Hopi, 38
- Hopi::PayloadBigFile, 50
 - ~PayloadBigFile, 50
- PayloadBigFile, 50
- Hopi::PayloadFile, 52
 - ~PayloadFile, 52
- PayloadFile, 52
- Host
 - gridftpd::LdapQuery, 46
- ID
 - ARex::ARexJob, 19
- Identity, 39
- Identity::Item, 40
- IdentityGACL, 39
- IdentityItemDN, 39
- IdentityItemVOMS, 40
- Index, 40
- ISIS::ISISService, 40
- ISIS::ISISecAttr, 40
- ISIS::Neighbor_Container, 47
- Janitor, 41
 - deploy, 42
 - Janitor, 42
 - remove, 42
 - result, 42
 - wait, 42
- job_state_rec_t, 43
- JobDescription, 43
- JobLocalDescription, 43
- JobLog, 43
- JobPlugin, 43
- Jobs
 - ARex::ARexJob, 19
- JobsList, 44
- JobsListConfig, 44
- JobUser, 44
- JobUserHelper, 45
- JobUsers, 45
- LdapQuery
 - gridftpd::LdapQuery, 46
- LdapQueryError
 - gridftpd::LdapQueryError, 47
- LogDir
 - ARex::ARexJob, 19
- LogFiles
 - ARex::ARexJob, 19
- logger
 - DREService::DREWebService, 31
- LRMSResult, 47
- makeFault

- DREService::DREWebService, 30
- nordugrid_config_loc
 - GMEEnvironment, 37
 - gridftpd::GMEEnvironment, 37
- ns_
 - DREService::DREWebService, 31
- numvalue_for_shell, 48
- ObjectAccess, 49
- ObjectAccess::Item, 41
- ObjectAccessGACL, 49
- OpenDir
 - ARex::ARexJob, 19
- OpenFile
 - ARex::ARexJob, 19
- OpenLogFile
 - ARex::ARexJob, 20
- operator bool
 - Cache::CacheService, 24
- parseINIConf
 - CacheConfig, 22
- PayloadBigFile
 - ARex::PayloadBigFile, 51
 - Hopi::PayloadBigFile, 50
- PayloadFile
 - ARex::PayloadFile, 51
 - Hopi::PayloadFile, 52
- PerlProcessor
 - DREService::PerlProcessor, 53
- Permission, 53
- PermissionGACL, 53
- Policy, 54
- process
 - Arc::Service_JavaWrapper, 56
 - Arc::Service_PythonWrapper, 56
 - Cache::CacheService, 24
 - DREService::DREWebService, 30
 - SPService::Service_SP, 58
- pushTask
 - DREService::TaskQueue, 59
- Query
 - gridftpd::LdapQuery, 46
- queryJobFinished
 - DTRGenerator, 32
- Read
 - ARex::ConfigIO, 27
 - ARex::NGConfig, 48
- ARex::XMLConfig, 63
- receiveDTR
 - DTRGenerator, 32
- receiveJob
 - DTRGenerator, 33
- RegistrationCollector
 - Cache::CacheService, 24
- Release
 - ARex::FileChunks, 34
- Remove
 - ARex::FileChunks, 34
- remove
 - Janitor, 42
- removeTask
 - DREService::TaskSet, 60
- Result
 - gridftpd::LdapQuery, 46
- result
 - Janitor, 42
- Resume
 - ARex::ARexJob, 20
- role
 - voms_attrs, 62
- RunFunction, 54
- RunParallel, 54
- RunPlugin, 54
- RunPlugin::lib_plugin_t, 47
- RunPlugins, 55
- RunRedirected, 55
- Scope
 - gridftpd::LdapQuery, 45
- Server, 55
- server
 - voms, 61
- Service_SP
 - SPService::Service_SP, 57
- SessionDir
 - ARex::ARexJob, 20
- setCacheDirs
 - CacheConfig, 22
- shiftTask
 - DREService::TaskQueue, 59
- SimpleMap, 58
- SPService::Service_SP, 57
 - process, 58
 - Service_SP, 57
- State
 - ARex::ARexJob, 20
- std

- voms, 61
- support_mail_address
 - GMEEnvironment, 37
 - gridftpd::GMEEnvironment, 37
- Task
 - DREService::Task, 58
- TaskQueue
 - DREService::TaskQueue, 59
- TaskSet
 - DREService::TaskSet, 59
- TotalJobs
 - ARex::ARexJob, 20
- UnixMap, 60
- UpdateCredentials
 - ARex::ARexJob, 20
- UrlMapConfig, 60
- userspec_t, 60
- value_for_shell, 61
- voms, 61
 - server, 61
 - std, 61
 - voname, 61
- voms_attrs, 61
 - cap, 62
 - group, 62
 - role, 62
- voname
 - voms, 61
- wait
 - Janitor, 42
- Write
 - ARex::ConfigIO, 27
 - ARex::NGConfig, 48
 - ARex::XMLConfig, 63
- ZeroUInt, 63