Hosting Environment (Daemon) Services

Generated by Doxygen 1.6.1

Wed May 5 18:33:38 2010

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Data Structure Index **3**

2.1 Class Hierarchy 3

3 Data Structure Index **5**

3.1 Data Structures 5

4 File Index **7**

4.1 File List 7

5 Namespace Documentation **9**

5.1 DREService Namespace Reference 9

5.1.1 Detailed Description 9

6 Data Structure Documentation **11**

6.1 ARex::ARexGMConfig Class Reference 11

6.2 ARex::ARexJob Class Reference 12

6.2.1 Detailed Description 12

6.2.2 Constructor & Destructor Documentation 12

6.2.2.1 ARexJob 12

6.2.2.2 ARexJob 13

6.2.3 Member Function Documentation 13

6.2.3.1 Cancel 13

6.2.3.2 ChooseSessionDir 13

6.2.3.3 Clean 13

6.2.3.4 CreateFile 13

6.2.3.5 Failed 13

6.2.3.6 Failure 13

6.2.3.7 GetDescription . . . 13
6.2.3.8 ID . . . 13
6.2.3.9 Jobs . . . 13
6.2.3.10 LogDir . . . 14
6.2.3.11 LogFiles . . . 14
6.2.3.12 OpenDir . . . 14
6.2.3.13 OpenFile . . . 14
6.2.3.14 OpenLogFile . . . 14
6.2.3.15 Resume . . . 14
6.2.3.16 SessionDir . . . 14
6.2.3.17 State . . . 14
6.2.3.18 State . . . 14
6.2.3.19 TotalJobs . . . 14
6.2.3.20 UpdateCredentials . . . 14
6.3 ARex::ARexService Class Reference . . . 16
6.4 CacheConfig Class Reference . . . 17
6.4.1 Detailed Description . . . 17
6.4.2 Constructor & Destructor Documentation . . . 17
6.4.2.1 CacheConfig . . . 17
6.4.3 Member Function Documentation . . . 17
6.4.3.1 parseINIConf . . . 17
6.4.3.2 setCacheDirs . . . 17
6.5 CacheConfigException Class Reference . . . 18
6.5.1 Detailed Description . . . 18
6.6 ArcSec::Charon Class Reference . . . 19
6.6.1 Detailed Description . . . 19
6.7 CommFIFO Class Reference . . . 20
6.8 ARex::ConfGrp Class Reference . . . 21
6.9 ARex::Config Class Reference . . . 22
6.9.1 Detailed Description . . . 22
6.9.2 Member Function Documentation . . . 22
6.9.2.1 ConfValue . . . 22
6.9.2.2 FirstConfValue . . . 22
6.9.2.3 GetConfigs . . . 22
6.10 ARex::ConfigError Class Reference . . . 23
6.10.1 Detailed Description . . . 23

6.10.2 Constructor & Destructor Documentation . . . 23
6.10.2.1 ConfigError . . . 23
6.11 ARex::ConfigIO Class Reference . . . 24
6.11.1 Detailed Description . . . 24
6.11.2 Member Function Documentation . . . 24
6.11.2.1 Read . . . 24
6.11.2.2 Write . . . 24
6.12 ConfigSections Class Reference . . . 25
6.13 ContinuationPlugins Class Reference . . . 26
6.14 ARex::CountedResource Class Reference . . . 27
6.15 Daemon Class Reference . . . 28
6.16 DREService::DREWebService Class Reference . . . 29
6.16.1 Constructor & Destructor Documentation . . . 29
6.16.1.1 DREWebService . . . 29
6.16.1.2 ∼DREWebService . . . 29
6.16.2 Member Function Documentation . . . 29
6.16.2.1 makeFault . . . 29
6.16.2.2 process . . . 30
6.16.3 Field Documentation . . . 30
6.16.3.1 logger . . . 30
6.16.3.2 ns_ . . . 30
6.17 ARex::FileChunks Class Reference . . . 31
6.17.1 Detailed Description . . . 31
6.17.2 Member Function Documentation . . . 31
6.17.2.1 Release . . . 31
6.17.2.2 Remove . . . 31
6.18 ARex::FileChunksList Class Reference . . . 32
6.18.1 Detailed Description . . . 32
6.18.2 Member Function Documentation . . . 32
6.18.2.1 Get . . . 32
6.18.2.2 GetStuck . . . 32
6.19 FileData Class Reference . . . 33
6.20 ARex::GridManager Class Reference . . . 34
6.21 Hopi::Hopi Class Reference . . . 35
6.22 ISIS::ISIService Class Reference . . . 36
6.23 ISIS::ISISSecAttr Class Reference . . . 37

6.24 Janitor Class Reference . . . 38
6.24.1 Detailed Description . . . 38
6.24.2 Constructor & Destructor Documentation . . . 38
6.24.2.1 Janitor . . . 38
6.24.3 Member Function Documentation . . . 38
6.24.3.1 deploy . . . 38
6.24.3.2 remove . . . 38
6.24.3.3 result . . . 38
6.24.3.4 wait . . . 39
6.25 job_state_rec_t Struct Reference . . . 40
6.26 JobDescription Class Reference . . . 41
6.27 JobLocalDescription Class Reference . . . 42
6.28 JobLog Class Reference . . . 43
6.28.1 Detailed Description . . . 43
6.29 ARex::JobRecord Class Reference . . . 44
6.30 JobsList Class Reference . . . 45
6.31 JobUser Class Reference . . . 46
6.32 JobUserHelper Class Reference . . . 47
6.33 JobUsers Class Reference . . . 48
6.34 RunPlugin::lib_plugin_t Union Reference . . . 49
6.35 ARex::LoggerClient Class Reference . . . 50
6.36 LRMSResult Class Reference . . . 51
6.37 ISIS::Neighbor_Container Class Reference . . . 52
6.38 ARex::NGConfig Class Reference . . . 53
6.38.1 Detailed Description . . . 53
6.38.2 Member Function Documentation . . . 53
6.38.2.1 Read . . . 53
6.38.2.2 Write . . . 53
6.39 numvalue_for_shell Class Reference . . . 54
6.40 ARex::OptimizedInformationContainer Class Reference . . . 55
6.41 ARex::Option Class Reference . . . 56
6.42 ARex::PayloadBigFile Class Reference . . . 57
6.42.1 Constructor & Destructor Documentation . . . 57
6.42.1.1 PayloadBigFile . . . 57
6.42.1.2 ∼PayloadBigFile . . . 57
6.43 Hopi::PayloadBigFile Class Reference . . . 58

6.43.1 Constructor & Destructor Documentation . . . 58
6.43.1.1 PayloadBigFile . . . 58
6.43.1.2 ~PayloadBigFile . . . 58
6.44 Hopi::PayloadFile Class Reference . . . 59
6.44.1 Detailed Description . . . 59
6.44.2 Constructor & Destructor Documentation . . . 59
6.44.2.1 PayloadFile . . . 59
6.44.2.2 ~PayloadFile . . . 59
6.45 ARex::PayloadFile Class Reference . . . 60
6.45.1 Detailed Description . . . 60
6.45.2 Constructor & Destructor Documentation . . . 60
6.45.2.1 PayloadFile . . . 60
6.45.2.2 ~PayloadFile . . . 60
6.46 DREService::PerlProcessor Class Reference . . . 61
6.46.1 Constructor & Destructor Documentation . . . 61
6.46.1.1 PerlProcessor . . . 61
6.46.1.2 ~PerlProcessor . . . 61
6.47 ArcSec::Charon::PolicyLocation Class Reference . . . 62
6.48 ContinuationPlugins::result_t Class Reference . . . 63
6.49 RunFunction Class Reference . . . 64
6.50 RunParallel Class Reference . . . 65
6.51 RunPlugin Class Reference . . . 66
6.52 RunPlugins Class Reference . . . 67
6.53 RunRedirected Class Reference . . . 68
6.54 Arc::Service_PythonWrapper Class Reference . . . 69
6.54.1 Member Function Documentation . . . 69
6.54.1.1 process . . . 69
6.55 DREService::Task Class Reference . . . 70
6.55.1 Constructor & Destructor Documentation . . . 70
6.55.1.1 Task . . . 70
6.55.1.2 ~Task . . . 70
6.56 DREService::TaskQueue Class Reference . . . 71
6.56.1 Constructor & Destructor Documentation . . . 71
6.56.1.1 TaskQueue . . . 71
6.56.1.2 ~TaskQueue . . . 71
6.56.2 Member Function Documentation . . . 71

6.56.2.1 pushTask . . . 71
6.56.2.2 shiftTask . . . 71
6.57 DREService::TaskSet Class Reference . . . 72
6.57.1 Constructor & Destructor Documentation . . . 72
6.57.1.1 TaskSet . . . 72
6.57.1.2 ∼TaskSet . . . 72
6.57.2 Member Function Documentation . . . 72
6.57.2.1 removeTask . . . 72
6.58 UrlMapConfig Class Reference . . . 73
6.59 value_for_shell Class Reference . . . 74
6.60 ARex::XMLConfig Class Reference . . . 75
6.60.1 Detailed Description . . . 75
6.60.2 Member Function Documentation . . . 75
6.60.2.1 Read . . . 75
6.60.2.2 Write . . . 75

7 File Documentation **77**
7.1 configcore.h File Reference . . . 77
7.1.1 Detailed Description . . . 77

Chapter 1

Namespace Index

1.1 Namespace List

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

DREService 9

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

ARex::ARexGMConfig . . . 11
ARex::ARexJob . . . 12
ARex::ARexService . . . 16
CacheConfig . . . 17
CacheConfigException . . . 18
ArcSec::Charon . . . 19
CommFIFO . . . 20
ARex::ConfGrp . . . 21
ARex::Config . . . 22
ARex::ConfigError . . . 23
ARex::ConfigIO . . . 24
    ARex::NGConfig . . . 53
    ARex::XMLConfig . . . 75
ConfigSections . . . 25
ContinuationPlugins . . . 26
ARex::CountedResource . . . 27
Daemon . . . 28
DREService::DREWebService . . . 29
ARex::FileChunks . . . 31
ARex::FileChunksList . . . 32
FileData . . . 33
ARex::GridManager . . . 34
Hopi::Hopi . . . 35
ISIS::ISIService . . . 36
ISIS::ISISSecAttr . . . 37
Janitor . . . 38
job_state_rec_t . . . 40
JobDescription . . . 41
JobLocalDescription . . . 42
JobLog . . . 43
ARex::JobRecord . . . 44
JobsList . . . 45
JobUser . . . 46

JobUserHelper 47
JobUsers 48
RunPlugin::lib_plugin_t 49
ARex::LoggerClient 50
LRMSResult 51
ISIS::Neighbor_Container 52
numvalue_for_shell 54
ARex::OptimizedInformationContainer 55
ARex::Option 56
ARex::PayloadBigFile 57
Hopi::PayloadBigFile 58
Hopi::PayloadFile 59
ARex::PayloadFile 60
DREService::PerlProcessor 61
ArcSec::Charon::PolicyLocation 62
ContinuationPlugins::result_t 63
RunFunction 64
RunParallel 65
RunPlugin 66
RunPlugins 67
RunRedirected 68
Arc::Service_PythonWrapper 69
DREService::Task 70
DREService::TaskQueue 71
DREService::TaskSet 72
UrlMapConfig 73
value_for_shell 74

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

ARex::ARexGMConfig . . . 11
ARex::ARexJob . . . 12
ARex::ARexService . . . 16
CacheConfig . . . 17
CacheConfigException . . . 18
ArcSec::Charon . . . 19
CommFIFO . . . 20
ARex::ConfGrp . . . 21
ARex::Config . . . 22
ARex::ConfigError . . . 23
ARex::ConfigIO . . . 24
ConfigSections . . . 25
ContinuationPlugins . . . 26
ARex::CountedResource . . . 27
Daemon . . . 28
DREService::DREWebService . . . 29
ARex::FileChunks (Representation of delivered file chunks) . . . 31
ARex::FileChunksList (Container for **FileChunks** (p. 31) instances) . . . 32
FileData . . . 33
ARex::GridManager . . . 34
Hopi::Hopi . . . 35
ISIS::ISIService . . . 36
ISIS::ISISSecAttr . . . 37
Janitor (Class to communicate with **Janitor** (p. 38) - Dynmaic Runtime Environment handler) . . . 38
job_state_rec_t . . . 40
JobDescription . . . 41
JobLocalDescription . . . 42
JobLog . . . 43
ARex::JobRecord . . . 44
JobsList . . . 45
JobUser . . . 46
JobUserHelper . . . 47
JobUsers . . . 48

RunPlugin::lib_plugin_t . . . 49
ARex::LoggerClient . . . 50
LRMSResult . . . 51
ISIS::Neighbor_Container . . . 52
ARex::NGConfig . . . 53
numvalue_for_shell . . . 54
ARex::OptimizedInformationContainer . . . 55
ARex::Option . . . 56
ARex::PayloadBigFile . . . 57
Hopi::PayloadBigFile . . . 58
Hopi::PayloadFile . . . 59
ARex::PayloadFile . . . 60
DREService::PerlProcessor . . . 61
ArcSec::Charon::PolicyLocation . . . 62
ContinuationPlugins::result_t . . . 63
RunFunction . . . 64
RunParallel . . . 65
RunPlugin . . . 66
RunPlugins . . . 67
RunRedirected . . . 68
Arc::Service_PythonWrapper . . . 69
DREService::Task . . . 70
DREService::TaskQueue . . . 71
DREService::TaskSet . . . 72
UrlMapConfig . . . 73
value_for_shell . . . 74
ARex::XMLConfig . . . 75

Chapter 4

File Index

4.1 File List

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

arex.h . . . ??
canonical_dir.h . . . ??
charon.h . . . ??
client.h . . . ??
commfifo.h . . . ??
conf.h . . . ??
conf_cache.h . . . ??
conf_file.h . . . ??
conf_map.h . . . ??
conf_pre.h . . . ??
conf_sections.h . . . ??
configcore.h . . . 77
configio.h . . . ??
daemon.h . . . ??
delete.h . . . ??
dREWebService.h . . . ??
environment.h . . . ??
escaped.h . . . ??
FileChunks.h . . . ??
gacl.h . . . ??
grid_manager.h . . . ??
gridmap.h . . . ??
hopi.h . . . ??
info_files.h . . . ??
info_log.h . . . ??
info_types.h . . . ??
isis.h . . . ??
janitor.h . . . ??
grid-manager/jobs/job.h . . . ??
job.h . . . ??
job_desc.h . . . ??
job_log.h . . . ??
job_request.h . . . ??

JobRecord.h . . . . . ??
LDIFtoXML.h . . . . . ??
ngconfig.h . . . . . ??
a-rex/PayloadFile.h . . . . . ??
hopi/PayloadFile.h . . . . . ??
PerlProcessor.h . . . . . ??
plugins.h . . . . . ??
proxy.h . . . . . ??
pythonwrapper.h . . . . . ??
run_function.h . . . . . ??
run_parallel.h . . . . . ??
run_plugin.h . . . . . ??
run_redirected.h . . . . . ??
security.h . . . . . ??
send_mail.h . . . . . ??
states.h . . . . . ??
Task.h . . . . . ??
TaskQueue.h . . . . . ??
TaskSet.h . . . . . ??
tools.h . . . . . ??
users.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 CacheConfig Class Reference

`#include <conf_cache.h>`

Public Member Functions

- **CacheConfig** (std::string username="")
- void **parseINIConf** (std::string username, **ConfigSections** *cf)
- void **setCacheDirs** (std::vector< std::string > cache_dirs)

6.4.1 Detailed Description

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

6.4.2 Constructor & Destructor Documentation

6.4.2.1 CacheConfig::CacheConfig (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.

6.4.3 Member Function Documentation

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

Parsers for the two different conf styles

6.4.3.2 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

6.5 CacheConfigException Class Reference

`#include <conf_cache.h>`

6.5.1 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

6.6 ArcSec::Charon Class Reference

`#include <charon.h>`

Data Structures

- class **PolicyLocation**

6.6.1 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

6.7 CommFIFO Class Reference

Data Structures

- class **elem_t**

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

- commfifo.h

6.8 ARex::ConfGrp Class Reference

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

- **configcore.h**

6.9 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

6.9.1 Detailed Description

Core configuration class.

6.9.2 Member Function Documentation

6.9.2.1 std::list<std::string> ARex::Config::ConfValue (const std::string & *path*) const

Get the configuration values from key.

6.9.2.2 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.

6.9.2.3 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**

6.10 ARex::ConfigError Class Reference

`#include <configcore.h>`

Public Member Functions

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

6.10.1 Detailed Description

Error configuration class.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 ARex::ConfigError::ConfigError (std::string *message*) [inline]

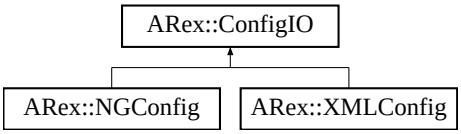
Constructor for the **ConfigError** (p. 23) exception. Calls the corresponding constructor in ARCLibError.

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

- **configcore.h**

6.11 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

6.11.1 Detailed Description

Virtual base-class for reading and writing configuration files. Concrete instances include **NGConfig** (p. 53) and **XMLConfig** (p. 75).

6.11.2 Member Function Documentation

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

Read the named configuration source.

Implemented in **ARex::NGConfig** (p. 53), and **ARex::XMLConfig** (p. 75).

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

Write configuration to named configuration destination.

Implemented in **ARex::NGConfig** (p. 53), and **ARex::XMLConfig** (p. 75).

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

- configio.h

6.12 ConfigSections Class Reference

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

- conf_sections.h

6.13 ContinuationPlugins Class Reference

Data Structures

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

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

- plugins.h

6.14 ARex::CountedResource Class Reference

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

- arex.h

6.15 Daemon Class Reference

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

- daemon.h

6.16 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**

6.16.1 Constructor & Destructor Documentation

6.16.1.1 DREService::DREWebService::DREWebService (Arc::Config ∗ *cfg*)

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

6.16.1.2 virtual DREService::DREWebService::∼DREWebService (void) [virtual]

Destructor.

6.16.2 Member Function Documentation

6.16.2.1 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:

outmsg outgoing message

Returns:

Status of the result achieved

6.16.2.2 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:

inmsg incoming message

inmsg outgoing message

Returns:

Status of the result achieved

6.16.3 Field Documentation

6.16.3.1 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.

6.16.3.2 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

6.17 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)

6.17.1 Detailed Description

Representation of delivered file chunks.

6.17.2 Member Function Documentation

6.17.2.1 void ARex::FileChunks::Release (void)

Release reference obtained through **FileChunksList::Get()** (p. 32) method. This operation may lead to destruction of FileChunk instance hence previously obtained refrence mus tnot be used.

6.17.2.2 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. 31) after whole file is delivered in order to free resources associated with **FileChunks** (p. 31) instance.

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

- FileChunks.h

6.18 ARex::FileChunksList Class Reference

Container for **FileChunks** (p. 31) instances.

`#include <FileChunks.h>`

Public Member Functions

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

6.18.1 Detailed Description

Container for **FileChunks** (p. 31) instances.

6.18.2 Member Function Documentation

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

Returns previously created **FileChunks** (p. 31) 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.

6.18.2.2 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

6.19 FileData Class Reference

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

- info_types.h

6.20 ARex::GridManager Class Reference

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

- grid_manager.h

6.21 Hopi::Hopi Class Reference

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

- hopi.h

6.22 ISIS::ISIService Class Reference

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

- isis.h

6.23 ISIS::ISISSecAttr Class Reference

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

- security.h

6.24 Janitor Class Reference

Class to communicate with **Janitor** (p. 38) - Dynmaic Runtime Environment handler.

`#include <janitor.h>`

Public Member Functions

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

6.24.1 Detailed Description

Class to communicate with **Janitor** (p. 38) - Dynmaic Runtime Environment handler.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 Janitor::Janitor (const std::string & *id*, const std::string & *cdir*)

Creates instance representing job entry in **Janitor** (p. 38) database. Takes id for job identifier and cdir for the control directory of A-Rex. constructor does not register job in the **Janitor** (p. 38). It only associates job with this instance.

6.24.3 Member Function Documentation

6.24.3.1 bool Janitor::deploy (void)

Registers associated job with **Janitor** (p. 38) and deploys dynamic RTEs. This operation is asynchronous. Returned true means **Janitor** (p. 38) will be contacted and deployemnt will start soon. For obtaining result of operation see methods **wait()** (p. 39) and **result()** (p. 38). During this operation janitor utility is called with command register and optionally deploy.

6.24.3.2 bool Janitor::remove (void)

Removes job from those handled by **Janitor** (p. 38) and releases associated RTEs. This operation is asynchronous. Returned true means **Janitor** (p. 38) will be contacted and removal will start soon. For obtaining result of operation see methods **wait()** (p. 39) and **result()** (p. 38). During this operation janitor utility is called with command remove.

6.24.3.3 Result Janitor::result (void)

Returns true if operation initiated by **deploy()** (p. 38) or **remove()** (p. 38) succeeded. It should be called after **wait()** (p. 39) returned true.

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

Wait till operation initiated by **deploy()** (p. 38) or **remove()** (p. 38) 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 immeaditely.

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

- janitor.h

6.25 job_state_rec_t Struct Reference

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

- grid-manager/jobs/job.h

6.26 JobDescription Class Reference

Friends

- class **JobsList**

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

- grid-manager/jobs/job.h

6.27 JobLocalDescription Class Reference

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

- info_types.h

6.28 JobLog Class Reference

`#include <job_log.h>`

6.28.1 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

6.29 ARex::JobRecord Class Reference

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

- JobRecord.h

6.30 JobsList Class Reference

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

- states.h

6.31 JobUser Class Reference

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

- users.h

6.32 JobUserHelper Class Reference

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

- users.h

6.33 JobUsers Class Reference

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

- users.h

6.34 RunPlugin::lib_plugin_t Union Reference

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

- run_plugin.h

6.35 ARex::LoggerClient Class Reference

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

- client.h

6.36 LRMSResult Class Reference

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

- info_types.h

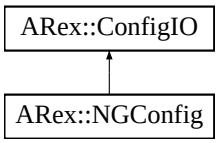
6.37 ISIS::Neighbor_Container Class Reference

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

- isis.h

6.38 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)

6.38.1 Detailed Description

Configuration class used for reading configuration files ARC-style.

6.38.2 Member Function Documentation

6.38.2.1 Config ARex::NGConfig::Read (std::istream & *is*) `[virtual]`

Read old arc.conf style configuration.

Implements **ARex::ConfigIO** (p. 24).

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

Write configuration to named file.

Implements **ARex::ConfigIO** (p. 24).

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

- ngconfig.h

6.39 numvalue_for_shell Class Reference

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

- job_desc.h

6.40 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

6.41 ARex::Option Class Reference

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

- **configcore.h**

6.42 ARex::PayloadBigFile Class Reference

Public Member Functions

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

6.42.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.42.1.2 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

6.43 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

6.43.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.43.1.2 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

6.44 Hopi::PayloadFile Class Reference

`#include <PayloadFile.h>`

Public Member Functions

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

6.44.1 Detailed Description

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

6.44.2 Constructor & Destructor Documentation

6.44.2.1 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.

6.44.2.2 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

6.45 ARex::PayloadFile Class Reference

`#include <PayloadFile.h>`

Public Member Functions

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

6.45.1 Detailed Description

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

6.45.2 Constructor & Destructor Documentation

6.45.2.1 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.

6.45.2.2 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

6.46 DREService::PerlProcessor Class Reference

Data Structures

- struct **ThreadInterface**

Public Member Functions

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

6.46.1 Constructor & Destructor Documentation

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

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

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

Destructor.

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

- PerlProcessor.h

6.47 ArcSec::Charon::PolicyLocation Class Reference

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

- charon.h

6.48 ContinuationPlugins::result_t Class Reference

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

- plugins.h

6.49 RunFunction Class Reference

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

- run_function.h

6.50 RunParallel Class Reference

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

- run_parallel.h

6.51 RunPlugin Class Reference

Data Structures

- union **lib_plugin_t**

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

- run_plugin.h

6.52 RunPlugins Class Reference

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

- run_plugin.h

6.53 RunRedirected Class Reference

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

- run_redirected.h

6.54 Arc::Service_PythonWrapper Class Reference

Public Member Functions

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

6.54.1 Member Function Documentation

6.54.1.1 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

6.55 DREService::Task Class Reference

Public Member Functions

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

6.55.1 Constructor & Destructor Documentation

6.55.1.1 DREService::Task::Task (int *taskID*, Arc::Message ∗ *request*, Arc::Message ∗ *response*)

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

6.55.1.2 virtual DREService::Task::∼Task (void) `[virtual]`

Destructor.

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

- Task.h

6.56 DREService::TaskQueue Class Reference

Public Member Functions

- **TaskQueue** (int length)
- virtual ∼**TaskQueue** (void)
- int **pushTask** (**Task** ∗task)
- **Task** ∗ **shiftTask** ()

6.56.1 Constructor & Destructor Documentation

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

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

6.56.1.2 virtual DREService::TaskQueue::∼TaskQueue (void) `[virtual]`

Destructor.

6.56.2 Member Function Documentation

6.56.2.1 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.

6.56.2.2 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

6.57 DREService::TaskSet Class Reference

Public Member Functions

- **TaskSet** (int size)
- virtual ∼**TaskSet** (void)
- **Task** ∗ **removeTask** (int)

6.57.1 Constructor & Destructor Documentation

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

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

6.57.1.2 virtual DREService::TaskSet::∼TaskSet (void) [virtual]

Destructor.

6.57.2 Member Function Documentation

6.57.2.1 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

6.58 UrlMapConfig Class Reference

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

- conf_map.h

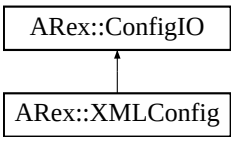
6.59 value_for_shell Class Reference

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

- job_desc.h

6.60 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)

6.60.1 Detailed Description

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

6.60.2 Member Function Documentation

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

Read configuration.

Implements **ARex::ConfigIO** (p. 24).

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

Write configuration.

Implements **ARex::ConfigIO** (p. 24).

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

- xmlconfig.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

~DREWebService
  DREService::DREWebService, 29
~PayloadBigFile
  ARex::PayloadBigFile, 57
  Hopi::PayloadBigFile, 58
~PayloadFile
  ARex::PayloadFile, 60
  Hopi::PayloadFile, 59
~PerlProcessor
  DREService::PerlProcessor, 61
~Task
  DREService::Task, 70
~TaskQueue
  DREService::TaskQueue, 71
~TaskSet
  DREService::TaskSet, 72

Arc::Service_PythonWrapper, 69
  process, 69
ArcSec::Charon, 19
ArcSec::Charon::PolicyLocation, 62
ARex::ARexGMConfig, 11
ARex::ARexJob, 12
  ARexJob, 12
  Cancel, 13
  ChooseSessionDir, 13
  Clean, 13
  CreateFile, 13
  Failed, 13
  Failure, 13
  GetDescription, 13
  ID, 13
  Jobs, 13
  LogDir, 13
  LogFiles, 14
  OpenDir, 14
  OpenFile, 14
  OpenLogFile, 14
  Resume, 14
  SessionDir, 14
  State, 14
  TotalJobs, 14
  UpdateCredentials, 14
ARex::ARexService, 16
ARex::ConfGrp, 21
ARex::Config, 22
  ConfValue, 22
  FirstConfValue, 22
  GetConfigs, 22
ARex::ConfigError, 23
  ConfigError, 23
ARex::ConfigIO, 24
  Read, 24
  Write, 24
ARex::CountedResource, 27
ARex::FileChunks, 31
  Release, 31
  Remove, 31
ARex::FileChunksList, 32
  Get, 32
  GetStuck, 32
ARex::GridManager, 34
ARex::JobRecord, 44
ARex::LoggerClient, 50
ARex::NGConfig, 53
  Read, 53
  Write, 53
ARex::OptimizedInformationContainer, 55
ARex::Option, 56
ARex::PayloadBigFile, 57
  ~PayloadBigFile, 57
  PayloadBigFile, 57
ARex::PayloadFile, 60
  ~PayloadFile, 60
  PayloadFile, 60
ARex::XMLConfig, 75
  Read, 75
  Write, 75
ARexJob
  ARex::ARexJob, 12

CacheConfig, 17
  CacheConfig, 17
  parseINIConf, 17
  setCacheDirs, 17
CacheConfigException, 18
Cancel
  ARex::ARexJob, 13
ChooseSessionDir
  ARex::ARexJob, 13

Clean
ARex::ARexJob, 13
CommFIFO, 20
configcore.h, 77
ConfigError
ARex::ConfigError, 23
ConfigSections, 25
ConfValue
ARex::Config, 22
ContinuationPlugins, 26
ContinuationPlugins::result_t, 63
CreateFile
ARex::ARexJob, 13

Daemon, 28
deploy
Janitor, 38
DREService, 9
DREService::DREWebService, 29
~DREWebService, 29
DREWebService, 29
logger, 30
makeFault, 29
ns_, 30
process, 29
DREService::PerlProcessor, 61
~PerlProcessor, 61
PerlProcessor, 61
DREService::Task, 70
~Task, 70
Task, 70
DREService::TaskQueue, 71
~TaskQueue, 71
pushTask, 71
shiftTask, 71
TaskQueue, 71
DREService::TaskSet, 72
~TaskSet, 72
removeTask, 72
TaskSet, 72
DREWebService
DREService::DREWebService, 29

Failed
ARex::ARexJob, 13
Failure
ARex::ARexJob, 13
FileData, 33
FirstConfValue
ARex::Config, 22

Get
ARex::FileChunksList, 32
GetConfigs
ARex::Config, 22
GetDescription
ARex::ARexJob, 13
GetStuck
ARex::FileChunksList, 32

Hopi::Hopi, 35
Hopi::PayloadBigFile, 58
~PayloadBigFile, 58
PayloadBigFile, 58
Hopi::PayloadFile, 59
~PayloadFile, 59
PayloadFile, 59

ID
ARex::ARexJob, 13
ISIS::ISIService, 36
ISIS::ISISSecAttr, 37
ISIS::Neighbor_Container, 52

Janitor, 38
deploy, 38
Janitor, 38
remove, 38
result, 38
wait, 38
job_state_rec_t, 40
JobDescription, 41
JobLocalDescription, 42
JobLog, 43
Jobs
ARex::ARexJob, 13
JobsList, 45
JobUser, 46
JobUserHelper, 47
JobUsers, 48

LogDir
ARex::ARexJob, 13
LogFiles
ARex::ARexJob, 14
logger
DREService::DREWebService, 30
LRMSResult, 51

makeFault
DREService::DREWebService, 29

ns_
DREService::DREWebService, 30
numvalue_for_shell, 54

OpenDir
ARex::ARexJob, 14
OpenFile

ARex::ARexJob, 14
OpenLogFile
ARex::ARexJob, 14

parseINIConf
CacheConfig, 17
PayloadBigFile
ARex::PayloadBigFile, 57
Hopi::PayloadBigFile, 58
PayloadFile
ARex::PayloadFile, 60
Hopi::PayloadFile, 59
PerlProcessor
DREService::PerlProcessor, 61
process
Arc::Service_PythonWrapper, 69
DREService::DREWebService, 29
pushTask
DREService::TaskQueue, 71

Read
ARex::ConfigIO, 24
ARex::NGConfig, 53
ARex::XMLConfig, 75
Release
ARex::FileChunks, 31
Remove
ARex::FileChunks, 31
remove
Janitor, 38
removeTask
DREService::TaskSet, 72
result
Janitor, 38
Resume
ARex::ARexJob, 14
RunFunction, 64
RunParallel, 65
RunPlugin, 66
RunPlugin::lib_plugin_t, 49
RunPlugins, 67
RunRedirected, 68

SessionDir
ARex::ARexJob, 14
setCacheDirs
CacheConfig, 17
shiftTask
DREService::TaskQueue, 71
State
ARex::ARexJob, 14

Task
DREService::Task, 70
TaskQueue
DREService::TaskQueue, 71
TaskSet
DREService::TaskSet, 72
TotalJobs
ARex::ARexJob, 14

UpdateCredentials
ARex::ARexJob, 14
UrlMapConfig, 73

value_for_shell, 74

wait
Janitor, 38
Write
ARex::ConfigIO, 24
ARex::NGConfig, 53
ARex::XMLConfig, 75