

Hosting Environment (Daemon) Services

Generated by Doxygen 1.6.1

Thu Nov 19 15:32:19 2009

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	Clean	13
6.2.3.3	CreateFile	13
6.2.3.4	Failed	13
6.2.3.5	Failure	13
6.2.3.6	GetDescription	13

6.2.3.7	ID	13
6.2.3.8	Jobs	13
6.2.3.9	LogDir	13
6.2.3.10	LogFiles	14
6.2.3.11	OpenDir	14
6.2.3.12	OpenFile	14
6.2.3.13	OpenLogFile	14
6.2.3.14	Resume	14
6.2.3.15	SessionDir	14
6.2.3.16	State	14
6.2.3.17	State	14
6.2.3.18	TotalJobs	14
6.2.3.19	UpdateCredentials	14
6.3	ARex::ARexService Class Reference	15
6.4	CacheConfig Class Reference	16
6.4.1	Detailed Description	16
6.4.2	Constructor & Destructor Documentation	16
6.4.2.1	CacheConfig	16
6.4.3	Member Function Documentation	16
6.4.3.1	parseINICnf	16
6.4.3.2	setCacheDirs	16
6.5	CacheConfigException Class Reference	17
6.5.1	Detailed Description	17
6.6	ArcSec::Charon Class Reference	18
6.6.1	Detailed Description	18
6.7	CommFIFO Class Reference	19
6.8	ARex::ConfGrp Class Reference	20
6.9	ARex::Config Class Reference	21
6.9.1	Detailed Description	21
6.9.2	Member Function Documentation	21
6.9.2.1	ConfValue	21
6.9.2.2	FirstConfValue	21
6.9.2.3	GetConfigs	21
6.10	ARex::ConfigError Class Reference	22
6.10.1	Detailed Description	22
6.10.2	Constructor & Destructor Documentation	22

6.10.2.1	ConfigError	22
6.11	ARex::ConfigIO Class Reference	23
6.11.1	Detailed Description	23
6.11.2	Member Function Documentation	23
6.11.2.1	Read	23
6.11.2.2	Write	23
6.12	ConfigSections Class Reference	24
6.13	ContinuationPlugins Class Reference	25
6.14	ARex::CountedResource Class Reference	26
6.15	Daemon Class Reference	27
6.16	DREService::DREWebService Class Reference	28
6.16.1	Constructor & Destructor Documentation	28
6.16.1.1	DREWebService	28
6.16.1.2	~DREWebService	28
6.16.2	Member Function Documentation	28
6.16.2.1	makeFault	28
6.16.2.2	process	29
6.16.3	Field Documentation	29
6.16.3.1	logger	29
6.16.3.2	ns_	29
6.17	FileData Class Reference	30
6.18	ARex::GridManager Class Reference	31
6.19	Hopi::Hopi Class Reference	32
6.20	ISIS::ISIService Class Reference	33
6.21	ISIS::ISISecAttr Class Reference	34
6.22	Janitor Class Reference	35
6.22.1	Detailed Description	35
6.22.2	Constructor & Destructor Documentation	35
6.22.2.1	Janitor	35
6.22.3	Member Function Documentation	35
6.22.3.1	deploy	35
6.22.3.2	remove	35
6.22.3.3	result	35
6.22.3.4	wait	36
6.23	job_state_rec_t Struct Reference	37
6.24	JobDescription Class Reference	38

6.25	JobLocalDescription Class Reference	39
6.26	JobLog Class Reference	40
6.26.1	Detailed Description	40
6.27	ARex::JobRecord Class Reference	41
6.28	JobsList Class Reference	42
6.29	JobUser Class Reference	43
6.30	JobUserHelper Class Reference	44
6.31	JobUsers Class Reference	45
6.32	RunPlugin::lib_plugin_t Union Reference	46
6.33	ARex::LoggerClient Class Reference	47
6.34	LRMSResult Class Reference	48
6.35	ARex::NGConfig Class Reference	49
6.35.1	Detailed Description	49
6.35.2	Member Function Documentation	49
6.35.2.1	Read	49
6.35.2.2	Write	49
6.36	numvalue_for_shell Class Reference	50
6.37	ARex::OptimizedInformationContainer Class Reference	51
6.38	ARex::Option Class Reference	52
6.39	Hopi::PayloadBigFile Class Reference	53
6.39.1	Constructor & Destructor Documentation	53
6.39.1.1	PayloadBigFile	53
6.39.1.2	~PayloadBigFile	53
6.40	ARex::PayloadBigFile Class Reference	54
6.40.1	Constructor & Destructor Documentation	54
6.40.1.1	PayloadBigFile	54
6.40.1.2	~PayloadBigFile	54
6.41	Hopi::PayloadFile Class Reference	55
6.41.1	Detailed Description	55
6.41.2	Constructor & Destructor Documentation	55
6.41.2.1	PayloadFile	55
6.41.2.2	~PayloadFile	55
6.42	ARex::PayloadFile Class Reference	56
6.42.1	Detailed Description	56
6.42.2	Constructor & Destructor Documentation	56
6.42.2.1	PayloadFile	56

6.42.2.2	~PayloadFile	56
6.43	DREService::PerlProcessor Class Reference	57
6.43.1	Constructor & Destructor Documentation	57
6.43.1.1	PerlProcessor	57
6.43.1.2	~PerlProcessor	57
6.44	RunFunction Class Reference	58
6.45	RunParallel Class Reference	59
6.46	RunPlugin Class Reference	60
6.47	RunPlugins Class Reference	61
6.48	RunRedirected Class Reference	62
6.49	Arc::Service_PythonWrapper Class Reference	63
6.49.1	Member Function Documentation	63
6.49.1.1	process	63
6.50	DREService::Task Class Reference	64
6.50.1	Constructor & Destructor Documentation	64
6.50.1.1	Task	64
6.50.1.2	~Task	64
6.51	DREService::TaskQueue Class Reference	65
6.51.1	Constructor & Destructor Documentation	65
6.51.1.1	TaskQueue	65
6.51.1.2	~TaskQueue	65
6.51.2	Member Function Documentation	65
6.51.2.1	pushTask	65
6.51.2.2	shiftTask	65
6.52	DREService::TaskSet Class Reference	66
6.52.1	Constructor & Destructor Documentation	66
6.52.1.1	TaskSet	66
6.52.1.2	~TaskSet	66
6.52.2	Member Function Documentation	66
6.52.2.1	removeTask	66
6.53	UrlMapConfig Class Reference	67
6.54	value_for_shell Class Reference	68
6.55	ARex::XMLConfig Class Reference	69
6.55.1	Detailed Description	69
6.55.2	Member Function Documentation	69
6.55.2.1	Read	69

6.55.2.2	Write	69
7	File Documentation	71
7.1	configcore.h File Reference	71
7.1.1	Detailed Description	71

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	15
CacheConfig	16
CacheConfigException	17
ArcSec::Charon	18
CommFIFO	19
ARex::ConfGrp	20
ARex::Config	21
ARex::ConfigError	22
ARex::ConfigIO	23
ARex::NGConfig	49
ARex::XMLConfig	69
ConfigSections	24
ContinuationPlugins	25
ARex::CountedResource	26
Daemon	27
DREService::DREWebService	28
FileData	30
ARex::GridManager	31
Hopi::Hopi	32
ISIS::ISISService	33
ISIS::ISISecAttr	34
Janitor	35
job_state_rec_t	37
JobDescription	38
JobLocalDescription	39
JobLog	40
ARex::JobRecord	41
JobsList	42
JobUser	43
JobUserHelper	44
JobUsers	45

RunPlugin::lib_plugin_t	46
ARex::LoggerClient	47
LRMSResult	48
numvalue_for_shell	50
ARex::OptimizedInformationContainer	51
ARex::Option	52
Hopi::PayloadBigFile	53
ARex::PayloadBigFile	54
Hopi::PayloadFile	55
ARex::PayloadFile	56
DREService::PerlProcessor	57
RunFunction	58
RunParallel	59
RunPlugin	60
RunPlugins	61
RunRedirected	62
Arc::Service_PythonWrapper	63
DREService::Task	64
DREService::TaskQueue	65
DREService::TaskSet	66
UrlMapConfig	67
value_for_shell	68

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	15
CacheConfig	16
CacheConfigException	17
ArcSec::Charon	18
CommFIFO	19
ARex::ConfGrp	20
ARex::Config	21
ARex::ConfigError	22
ARex::ConfigIO	23
ConfigSections	24
ContinuationPlugins	25
ARex::CountedResource	26
Daemon	27
DREService::DREWebService	28
FileData	30
ARex::GridManager	31
Hopi::Hopi	32
ISIS::ISISService	33
ISIS::ISISSecAttr	34
Janitor (Class to communicate with Janitor (p. 35) - Dynmaic Runtime Environment handler)	35
job_state_rec_t	37
JobDescription	38
JobLocalDescription	39
JobLog	40
ARex::JobRecord	41
JobsList	42
JobUser	43
JobUserHelper	44
JobUsers	45
RunPlugin::lib_plugin_t	46
ARex::LoggerClient	47

LRMSResult	48
ARex::NGConfig	49
numvalue_for_shell	50
ARex::OptimizedInformationContainer	51
ARex::Option	52
Hopi::PayloadBigFile	53
ARex::PayloadBigFile	54
Hopi::PayloadFile	55
ARex::PayloadFile	56
DREService::PerlProcessor	57
RunFunction	58
RunParallel	59
RunPlugin	60
RunPlugins	61
RunRedirected	62
Arc::Service_PythonWrapper	63
DREService::Task	64
DREService::TaskQueue	65
DREService::TaskSet	66
UrlMapConfig	67
value_for_shell	68
ARex::XMLConfig	69

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	71
configio.h	??
daemon.h	??
delete.h	??
dREWebService.h	??
environment.h	??
escaped.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, const 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)

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, const 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::Clean (void)`

Remove job from local pool

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

Creates file in job's session directory and returns handler

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

Returns true if job has failed

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

Returns textual description of failure of last operation

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

Fills provided jsdl with job description

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

Return ID assigned to job

6.2.3.8 `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.

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

Returns name of virtual log directory

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

Returns list of existing log files

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

Opens directory inside session directory

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

Opens file in job's session directory and returns handler

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

Opens log file in control directory

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

Resume execution of job after error

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

Returns path to session directory

6.2.3.16 `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

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

Returns current state of job

6.2.3.18 `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.

6.2.3.19 `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::list< 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. 16) 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::list< 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>
```

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]

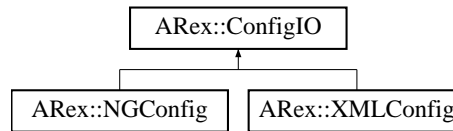
Constructor for the **ConfigError** (p. 22) exception. Calls the corresponding constructor in ARCLibError.

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. 49) and **XMLConfig** (p. 69).

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. 49), and **ARex::XMLConfig** (p. 69).

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. 49), and **ARex::XMLConfig** (p. 69).

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

- struct **command_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 FileData Class Reference

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

- `info_types.h`

6.18 ARex::GridManager Class Reference

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

- grid_manager.h

6.19 Hopi::Hopi Class Reference

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

- `hopi.h`

6.20 ISIS::ISIService Class Reference

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

- isis.h

6.21 ISIS::ISISecAttr Class Reference

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

- security.h

6.22 Janitor Class Reference

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

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

Public Member Functions

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

6.22.1 Detailed Description

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

6.22.2 Constructor & Destructor Documentation

6.22.2.1 Janitor::Janitor (const std::string &id, const std::string &cdir)

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

6.22.3 Member Function Documentation

6.22.3.1 bool Janitor::deploy (void)

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

6.22.3.2 bool Janitor::remove (void)

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

6.22.3.3 Result Janitor::result (void)

Returns true if operation initiated by **deploy()** (p. 35) or **remove()** (p. 35) succeeded. It should be called after **wait()** (p. 36) returned true.

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

Wait till operation initiated by **deploy()** (p. 35) or **remove()** (p. 35) 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

6.23 job_state_rec_t Struct Reference

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

- grid-manager/jobs/job.h

6.24 JobDescription Class Reference

Friends

- class **JobsList**

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

- grid-manager/jobs/job.h

6.25 JobLocalDescription Class Reference

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

- info_types.h

6.26 JobLog Class Reference

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

6.26.1 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

6.27 ARex::JobRecord Class Reference

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

- JobRecord.h

6.28 JobsList Class Reference

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

- `states.h`

6.29 JobUser Class Reference

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

- users.h

6.30 JobUserHelper Class Reference

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

- users.h

6.31 JobUsers Class Reference

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

- users.h

6.32 RunPlugin::lib_plugin_t Union Reference

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

- run_plugin.h

6.33 ARex::LoggerClient Class Reference

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

- client.h

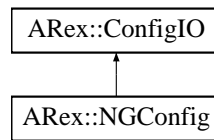
6.34 LRMSResult Class Reference

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

- info_types.h

6.35 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)

6.35.1 Detailed Description

Configuration class used for reading configuration files ARC-style.

6.35.2 Member Function Documentation

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

Read old arc.conf style configuration.

Implements **ARex::ConfigIO** (p. 23).

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

Write configuration to named file.

Implements **ARex::ConfigIO** (p. 23).

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

- ngconfig.h

6.36 numvalue_for_shell Class Reference

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

- `job_desc.h`

6.37 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

6.38 ARex::Option Class Reference

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

- **configcore.h**

6.39 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

6.39.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.39.1.2 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

6.40 ARex::PayloadBigFile Class Reference

Public Member Functions

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

6.40.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.40.1.2 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

6.41 Hopi::PayloadFile Class Reference

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

Public Member Functions

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

6.41.1 Detailed Description

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

6.41.2 Constructor & Destructor Documentation

6.41.2.1 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.

6.41.2.2 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

6.42 ARex::PayloadFile Class Reference

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

Public Member Functions

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

6.42.1 Detailed Description

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

6.42.2 Constructor & Destructor Documentation

6.42.2.1 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.

6.42.2.2 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

6.43 DREService::PerlProcessor Class Reference

Data Structures

- struct **ThreadInterface**

Public Member Functions

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

6.43.1 Constructor & Destructor Documentation

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

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

6.43.1.2 virtual DREService::PerlProcessor::~~PerlProcessor (void) [**virtual**]

Destructor.

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

- PerlProcessor.h

6.44 RunFunction Class Reference

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

- `run_function.h`

6.45 RunParallel Class Reference

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

- run_parallel.h

6.46 RunPlugin Class Reference

Data Structures

- union `lib_plugin_t`

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

- `run_plugin.h`

6.47 RunPlugins Class Reference

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

- run_plugin.h

6.48 RunRedirected Class Reference

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

- `run_redirected.h`

6.49 Arc::Service_PythonWrapper Class Reference

Public Member Functions

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

6.49.1 Member Function Documentation

6.49.1.1 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

6.50 DREService::Task Class Reference

Public Member Functions

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

6.50.1 Constructor & Destructor Documentation

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

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

6.50.1.2 virtual DREService::Task::~~Task (void) **[virtual]**

Destructor.

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

- Task.h

6.51 DREService::TaskQueue Class Reference

Public Member Functions

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

6.51.1 Constructor & Destructor Documentation

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

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

6.51.1.2 virtual DREService::TaskQueue::~~TaskQueue (void) **[virtual]**

Destructor.

6.51.2 Member Function Documentation

6.51.2.1 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.

6.51.2.2 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

6.52 DREService::TaskSet Class Reference

Public Member Functions

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

6.52.1 Constructor & Destructor Documentation

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

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

6.52.1.2 virtual DREService::TaskSet::~~TaskSet (void) **[virtual]**

Destructor.

6.52.2 Member Function Documentation

6.52.2.1 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

6.53 UrlMapConfig Class Reference

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

- `conf_map.h`

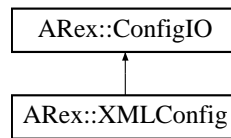
6.54 value_for_shell Class Reference

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

- job_desc.h

6.55 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)

6.55.1 Detailed Description

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

6.55.2 Member Function Documentation

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

Read configuration.

Implements **ARex::ConfigIO** (p. 23).

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

Write configuration.

Implements **ARex::ConfigIO** (p. 23).

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, 28
- ~PayloadBigFile
 - ARex::PayloadBigFile, 54
 - Hopi::PayloadBigFile, 53
- ~PayloadFile
 - ARex::PayloadFile, 56
 - Hopi::PayloadFile, 55
- ~PerlProcessor
 - DREService::PerlProcessor, 57
- ~Task
 - DREService::Task, 64
- ~TaskQueue
 - DREService::TaskQueue, 65
- ~TaskSet
 - DREService::TaskSet, 66
- Arc::Service_PythonWrapper, 63
 - process, 63
- ArcSec::Charon, 18
- ARex::ARexGMConfig, 11
- ARex::ARexJob, 12
 - ARexJob, 12
 - Cancel, 13
 - Clean, 13
 - CreateFile, 13
 - Failed, 13
 - Failure, 13
 - GetDescription, 13
 - ID, 13
 - Jobs, 13
 - LogDir, 13
 - LogFiles, 13
 - OpenDir, 14
 - OpenFile, 14
 - OpenLogFile, 14
 - Resume, 14
 - SessionDir, 14
 - State, 14
 - TotalJobs, 14
 - UpdateCredentials, 14
- ARex::ARexService, 15
- ARex::ConfGrp, 20
- ARex::Config, 21
 - ConfValue, 21
 - FirstConfValue, 21
 - GetConfigs, 21
- ARex::ConfigError, 22
 - ConfigError, 22
- ARex::ConfigIO, 23
 - Read, 23
 - Write, 23
- ARex::CountedResource, 26
- ARex::GridManager, 31
- ARex::JobRecord, 41
- ARex::LoggerClient, 47
- ARex::NGConfig, 49
 - Read, 49
 - Write, 49
- ARex::OptimizedInformationContainer, 51
- ARex::Option, 52
- ARex::PayloadBigFile, 54
 - ~PayloadBigFile, 54
 - PayloadBigFile, 54
- ARex::PayloadFile, 56
 - ~PayloadFile, 56
 - PayloadFile, 56
- ARex::XMLConfig, 69
 - Read, 69
 - Write, 69
- ARexJob
 - ARex::ARexJob, 12
- CacheConfig, 16
 - CacheConfig, 16
 - parseINIConf, 16
 - setCacheDirs, 16
- CacheConfigException, 17
- Cancel
 - ARex::ARexJob, 13
- Clean
 - ARex::ARexJob, 13
- CommFIFO, 19
- configcore.h, 71
- ConfigError
 - ARex::ConfigError, 22
- ConfigSections, 24
- ConfValue
 - ARex::Config, 21
- ContinuationPlugins, 25

- CreateFile
 - ARex::ARexJob, 13
- Daemon, 27
- deploy
 - Janitor, 35
- DREService, 9
- DREService::DREWebService, 28
 - ~DREWebService, 28
 - DREWebService, 28
 - logger, 29
 - makeFault, 28
 - ns_, 29
 - process, 28
- DREService::PerlProcessor, 57
 - ~PerlProcessor, 57
 - PerlProcessor, 57
- DREService::Task, 64
 - ~Task, 64
 - Task, 64
- DREService::TaskQueue, 65
 - ~TaskQueue, 65
 - pushTask, 65
 - shiftTask, 65
 - TaskQueue, 65
- DREService::TaskSet, 66
 - ~TaskSet, 66
 - removeTask, 66
 - TaskSet, 66
- DREWebService
 - DREService::DREWebService, 28
- Failed
 - ARex::ARexJob, 13
- Failure
 - ARex::ARexJob, 13
- FileData, 30
- FirstConfValue
 - ARex::Config, 21
- GetConfigs
 - ARex::Config, 21
- GetDescription
 - ARex::ARexJob, 13
- Hopi::Hopi, 32
- Hopi::PayloadBigFile, 53
 - ~PayloadBigFile, 53
 - PayloadBigFile, 53
- Hopi::PayloadFile, 55
 - ~PayloadFile, 55
 - PayloadFile, 55
- ID
 - ARex::ARexJob, 13
- ISIS::ISIService, 33
- ISIS::ISISecAttr, 34
- Janitor, 35
 - deploy, 35
 - Janitor, 35
 - remove, 35
 - result, 35
 - wait, 35
- job_state_rec_t, 37
- JobDescription, 38
- JobLocalDescription, 39
- JobLog, 40
- Jobs
 - ARex::ARexJob, 13
- JobsList, 42
- JobUser, 43
- JobUserHelper, 44
- JobUsers, 45
- LogDir
 - ARex::ARexJob, 13
- LogFiles
 - ARex::ARexJob, 13
- logger
 - DREService::DREWebService, 29
- LRMSResult, 48
- makeFault
 - DREService::DREWebService, 28
- ns_
 - DREService::DREWebService, 29
- numvalue_for_shell, 50
- OpenDir
 - ARex::ARexJob, 14
- OpenFile
 - ARex::ARexJob, 14
- OpenLogFile
 - ARex::ARexJob, 14
- parseINICnf
 - CacheConfig, 16
- PayloadBigFile
 - ARex::PayloadBigFile, 54
 - Hopi::PayloadBigFile, 53
- PayloadFile
 - ARex::PayloadFile, 56
 - Hopi::PayloadFile, 55
- PerlProcessor
 - DREService::PerlProcessor, 57
- process
 - Arc::Service_PythonWrapper, 63
 - DREService::DREWebService, 28

- pushTask
 - DREService::TaskQueue, 65
- Read
 - ARex::ConfigIO, 23
 - ARex::NGConfig, 49
 - ARex::XMLConfig, 69
- remove
 - Janitor, 35
- removeTask
 - DREService::TaskSet, 66
- result
 - Janitor, 35
- Resume
 - ARex::ARexJob, 14
- RunFunction, 58
- RunParallel, 59
- RunPlugin, 60
- RunPlugin::lib_plugin_t, 46
- RunPlugins, 61
- RunRedirected, 62
- SessionDir
 - ARex::ARexJob, 14
- setCacheDirs
 - CacheConfig, 16
- shiftTask
 - DREService::TaskQueue, 65
- State
 - ARex::ARexJob, 14
- Task
 - DREService::Task, 64
- TaskQueue
 - DREService::TaskQueue, 65
- TaskSet
 - DREService::TaskSet, 66
- TotalJobs
 - ARex::ARexJob, 14
- UpdateCredentials
 - ARex::ARexJob, 14
- UrlMapConfig, 67
- value_for_shell, 68
- wait
 - Janitor, 35
- Write
 - ARex::ConfigIO, 23
 - ARex::NGConfig, 49
 - ARex::XMLConfig, 69