

Hosting Environment (Daemon) Services Reference Manual

Generated by Doxygen 1.4.7

Sun Aug 31 00:50:03 2008

Contents

1	Hosting Environment (Daemon) Services Hierarchical Index	1
1.1	Hosting Environment (Daemon) Services Class Hierarchy	1
2	Hosting Environment (Daemon) Services Class Index	3
2.1	Hosting Environment (Daemon) Services Class List	3
3	Hosting Environment (Daemon) Services File Index	5
3.1	Hosting Environment (Daemon) Services File List	5
4	Hosting Environment (Daemon) Services Class Documentation	9
4.1	ARCLibError Class Reference	9
4.2	ARex::ARexJob Class Reference	11
4.3	ARex::Config Class Reference	13
4.4	ARex::ConfigError Class Reference	14
4.5	ARex::ConfigIO Class Reference	15
4.6	Job Class Reference	16
4.7	ARex2::JobControl Class Reference	18
4.8	ARex2::JobDataCache Class Reference	19
4.9	ARex2::JobDescription Class Reference	20
4.10	ARex2::JobDescription::InputFile Class Reference	22
4.11	ARex2::JobDescription::Notification Class Reference	23
4.12	ARex2::JobDescription::OutputFile Class Reference	24
4.13	JobLog Class Reference	25
4.14	ARex2::JobLRMSInfo Class Reference	26
4.15	JobRequestError Class Reference	27
4.16	JobRequestJSDL Class Reference	28
4.17	JobRequestXRSL Class Reference	29
4.18	ARex2::JobState Class Reference	30
4.19	ARex2::JobUser Class Reference	31

4.20	ARex::NGConfig Class Reference	32
4.21	HTTPD::PayloadFile Class Reference	33
4.22	ARex::PayloadFile Class Reference	34
4.23	RTE Class Reference	35
4.24	RuntimeEnvironment Class Reference	37
4.25	RuntimeEnvironmentError Class Reference	39
4.26	ArcSec::Service_AA Class Reference	40
4.27	ArcSec::Service_PDP Class Reference	41
4.28	Time Class Reference	42
4.29	TimeError Class Reference	45
4.30	ARex::XMLConfig Class Reference	46
4.31	Xrsl Class Reference	47
4.32	XrslError Class Reference	50
4.33	XrslRelation Class Reference	51
4.34	XrslValidationData Class Reference	53
5	Hosting Environment (Daemon) Services File Documentation	55
5.1	configcore.h File Reference	55

Chapter 1

Hosting Environment (Daemon) Services Hierarchical Index

1.1 Hosting Environment (Daemon) Services Class Hierarchy

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

ARCLibError	9
JobRequestError	27
RuntimeEnvironmentError	39
TimeError	45
XrslError	50
ARex::ARexJob	11
ARex::Config	13
ARex::ConfigError	14
ARex::ConfigIO	15
ARex::NGConfig	32
ARex::XMLConfig	46
Job	16
ARex2::JobControl	18
ARex2::JobDataCache	19
ARex2::JobDescription	20
ARex2::JobDescription::InputFile	22
ARex2::JobDescription::Notification	23
ARex2::JobDescription::OutputFile	24
JobLog	25
ARex2::JobLRMSInfo	26
JobRequestJSDL	28
JobRequestXRSL	29
ARex2::JobState	30
ARex2::JobUser	31
HTTPD::PayloadFile	33
ARex::PayloadFile	34
RTE	35
RuntimeEnvironment	37
ArcSec::Service_AA	40
ArcSec::Service_PDP	41

Time	42
Xrsl	47
XrslRelation	51
XrslValidationData	53

Chapter 2

Hosting Environment (Daemon) Services Class Index

2.1 Hosting Environment (Daemon) Services Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ARCLibError	9
ARex::ARexJob	11
ARex::Config	13
ARex::ConfigError	14
ARex::ConfigIO	15
Job	16
ARex2::JobControl	18
ARex2::JobDataCache	19
ARex2::JobDescription	20
ARex2::JobDescription::InputFile	22
ARex2::JobDescription::Notification	23
ARex2::JobDescription::OutputFile	24
JobLog	25
ARex2::JobLRMSInfo	26
JobRequestError	27
JobRequestJSDL	28
JobRequestXRSL	29
ARex2::JobState	30
ARex2::JobUser	31
ARex::NGConfig	32
HTTPD::PayloadFile	33
ARex::PayloadFile	34
RTE	35
RuntimeEnvironment	37
RuntimeEnvironmentError	39
ArcSec::Service_AA	40
ArcSec::Service_PDP	41
Time	42
TimeError	45
ARex::XMLConfig	46
Xrsl	47

XrslError	50
XrslRelation	51
XrslValidationData	53

Chapter 3

Hosting Environment (Daemon) Services File Index

3.1 Hosting Environment (Daemon) Services File List

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

aaservice.h	??
arex.h	??
arex2.h	??
cache.h	??
cache_cleaner.h	??
canonical_dir.h	??
commfifo.h	??
conf.h	??
conf_file.h	??
conf_map.h	??
conf_pre.h	??
conf_sections.h	??
configcore.h	55
configio.h	??
configurator.h	??
daemon.h	??
datetime.h	??
delete.h	??
environment.h	??
error.h	??
escaped.h	??
fsusage.h	??
gacl.h	??
grid_manager.h	??
grid_sched.h	??
gridmap.h	??
httpd.h	??
info_files.h	??
info_log.h	??
info_types.h	??
isis.h	??

javawrapper.h	??
a-rex/grid-manager/jobdesc/job.h	??
a-rex/grid-manager/jobs/job.h	??
a-rex/job.h	??
arex2/job.h	??
paul/job.h	??
sched/job.h	??
job_control.h	??
job_data_cache.h	??
job_descr.h	??
job_jsdl.h	??
job_list.h	??
job_log.h	??
paul/job_queue.h	??
sched/job_queue.h	??
a-rex/grid-manager/jobs/job_request.h	??
paul/job_request.h	??
sched/job_request.h	??
paul/job_sched_meta.h	??
sched/job_sched_meta.h	??
job_state.h	??
paul/job_status.h	??
sched/job_status.h	??
job_user.h	??
job_xrsl.h	??
jobdesc_util.h	??
jsdl_job.h	??
LDIFtoXML.h	??
lrms.h	??
ngconfig.h	??
parse_rsl.h	??
paul.h	??
a-rex/PayloadFile.h	??
httpd/PayloadFile.h	??
pdpSERVICE.h	??
plugins.h	??
proxy.h	??
pythonwrapper.h	??
resource.h	??
resources_handling.h	??
router.h	??
rte.h	??
run.h	??
run_function.h	??
run_parallel.h	??
run_plugin.h	??
run_redirected.h	??
runtimeenvironment.h	??
send_mail.h	??
states.h	??
stlvector.h	??
subst_rsl.h	??
sysinfo.h	??
tools.h	??

users.h	..	??
xmlconfig.h	..	??
xrsl.h	..	??

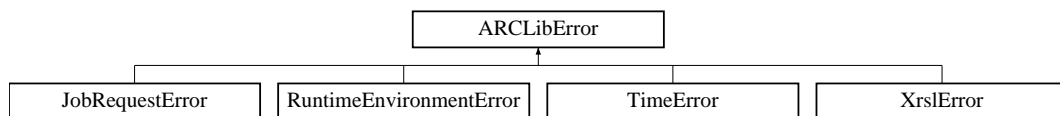
Chapter 4

Hosting Environment (Daemon) Services Class Documentation

4.1 ARCLibError Class Reference

```
#include <error.h>
```

Inheritance diagram for ARCLibError::



Public Member Functions

- [ARCLibError](#) (std::string message)
- virtual [~ARCLibError](#) () throw ()
- virtual const char * [what](#) () const throw ()

4.1.1 Detailed Description

This is the top exception for ARCLib. Every exception in ARCLib should inherit from this. The exception inherits from the top C++ exception: std::exception.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 ARCLibError::ARCLibError (std::string *message*) [inline]

Creates a new exception, with the message given as argument

4.1.2.2 virtual ARCLibError::~~ARCLibError () throw () [inline, virtual]

Destructor. Not that much to say.

4.1.3 Member Function Documentation

4.1.3.1 `virtual const char* ARCLibError::what (void) const throw ()` [inline, virtual]

Returns the message given in the constructor.

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

- error.h

4.2 ARex::ARexJob Class Reference

```
#include <job.h>
```

Public Member Functions

- [ARexJob](#) (const std::string &id, ARexGMConfig &config)
- [ARexJob](#) (Arc::XMLNode jsdl, ARexGMConfig &config, const std::string &credentials, const std::string &clientid)
- **operator bool** (void)
- bool **operator!** (void)
- 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)
- int [CreateFile](#) (const std::string &filename)

Static Public Member Functions

- static int [TotalJobs](#) (ARexGMConfig &config)
- static std::list< std::string > [Jobs](#) (ARexGMConfig &config)

4.2.1 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.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ARex::ARexJob::ARexJob (const std::string &id, ARexGMConfig &config)

Create instance which is an interface to existing job

4.2.2.2 ARex::ARexJob::ARexJob (Arc::XMLNode jsdl, ARexGMConfig &config, const std::string &credentials, const std::string &clientid)

Create new job with provided JSDL description

4.2.3 Member Function Documentation

4.2.3.1 bool ARex::ARexJob::Cancel (void)

Cancel processing/execution of job

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

Remove job from local pool

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

Returns true if job has failed

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

Returns textual description of failure of last operation

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

Fills provided jsdl with job description

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

Return ID assigned to job

4.2.3.7 bool ARex::ARexJob::Resume (void)

Resume execution of job after error

4.2.3.8 std::string ARex::ARexJob::SessionDir (void)

Returns path to session directory

4.2.3.9 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

4.2.3.10 std::string ARex::ARexJob::State (void)

Returns current state of job

4.2.3.11 static int ARex::ARexJob::TotalJobs (ARexGMConfig & config) [static]

Return number of jobs associated with this configuration. TODO: total for all user configurations.

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

- a-rex/job.h

4.3 ARex::Config Class Reference

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

Public Member Functions

- void **AddConfGrp** (const ConfGrp &confgrp)
- const ConfGrp & **FindConfGrp** (const std::string §ion, const std::string &id) const
- const std::list< ConfGrp > & **GetConfigs** () const
- std::list< std::string > **ConfValue** (const std::string &path) const
- std::string **FirstConfValue** (const std::string &path) const

4.3.1 Detailed Description

Core configuration class.

4.3.2 Member Function Documentation

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

Get the configuration values from key.

4.3.2.2 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.

4.3.2.3 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](#)

4.4 ARex::ConfigError Class Reference

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

Public Member Functions

- [ConfigError](#) (std::string message)
- std::string & **what** (void)

4.4.1 Detailed Description

Error configuration class.

4.4.2 Constructor & Destructor Documentation

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

Constructor for the [ConfigError](#) exception. Calls the corresponding constructor in [ARCLibError](#).

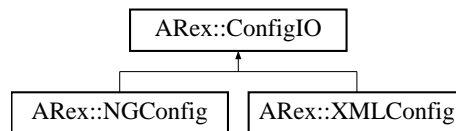
The documentation for this class was generated from the following file:

- [configcore.h](#)

4.5 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

4.5.1 Detailed Description

Virtual base-class for reading and writing configuration files. Concrete instances include [NGConfig](#) and [XMLConfig](#).

4.5.2 Member Function Documentation

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

Read the named configuration source.

Implemented in [ARex::NGConfig](#), and [ARex::XMLConfig](#).

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

Write configuration to named configuration destination.

Implemented in [ARex::NGConfig](#), and [ARex::XMLConfig](#).

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

- configio.h

4.6 Job Class Reference

```
#include <job.h>
```

Public Member Functions

- [Job](#) (void)
- [Job](#) (std::string path)
- [~Job](#) (void)
- [operator bool](#) (void)
- bool [operator!](#) (void)
- bool [Start](#) (void)
- bool [Cancel](#) (void)
- bool [Resume](#) (void)
- std::string [GetState](#) (void)
- std::string [GetSessionDir](#) (void)

4.6.1 Detailed Description

Collect all information (status, lrms info, user) required to handle job

4.6.2 Constructor & Destructor Documentation

4.6.2.1 [Job::Job](#) (void)

Constructor: Creates empty job

4.6.2.2 [Job::Job](#) (std::string *path*)

Constructor: load job information form files

4.6.2.3 [Job::~~Job](#) (void)

Destuctionior

4.6.3 Member Function Documentation

4.6.3.1 [bool Job::Cancel](#) (void)

Cancel processing/execution of job

4.6.3.2 [std::string Job::GetSessionDir](#) (void)

Returns the session directory of the job

4.6.3.3 std::string Job::GetState (void)

Returns the string represnetation of job state

4.6.3.4 Job::operator bool (void) [inline]

Helper logical operators

4.6.3.5 bool Job::Resume (void)

Resume execution of job after error

4.6.3.6 bool Job::Start (void)

Starts job. Most of the cases it means to submit to LRMS

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

- arex2/job.h

4.7 ARex2::JobControl Class Reference

```
#include <job_control.h>
```

Public Member Functions

- **JobControl** ([JobUser](#) &user, std::string &job_id)
- void **SetControlDir** (const std::string &dir)
- void **SetSessionDir** (const std::string &dir)
- const std::string & **ControlDir** (void) const
- const std::string & **SessionDir** (void) const
- bool **Serialize** ([JobState](#) &state)
- bool **Serialize** ([JobDescription](#) &desc)
- bool **Serialize** ([JobUser](#) &user)
- bool **Serialize** ([Job](#) &job)
- bool **DeSerialize** ([JobState](#) &state)
- bool **DeSerialize** ([JobDescription](#) &desc)
- bool **DeSerialize** ([JobUser](#) &user)
- bool **DeSerialize** ([Job](#) &job)

Protected Attributes

- std::string **control_dir_**
- std::string **session_dir_**

4.7.1 Detailed Description

Represents job controll information like session dir, control dir

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

- job_control.h

4.8 ARex2::JobDataCache Class Reference

```
#include <job_data_cache.h>
```

Public Member Functions

- **JobDataCache** (const std::string &dir, const std::string &data_dir, bool priv=false)
- **JobDataCache** (const std::string &dir, const std::string &data_dir, const std::string &link_dir, bool priv=false)
- void **SetCacheSize** (long long int cache_max, long long int cache_min=0)
- const std::string & **CacheDir** (void) const
- const std::string & **CacheDataDir** (void) const
- const std::string & **CacheLinkDir** (void) const
- long long int **CacheMaxSize** (void) const
- long long int **CacheMinSize** (void) const
- bool **CachePrivate** (void) const

4.8.1 Detailed Description

Data cache

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

- job_data_cache.h

4.9 ARex2::JobDescription Class Reference

```
#include <job_descr.h>
```

Public Member Functions

- **JobDescription** (const [JobDescription](#) &j)
- virtual [JobDescription](#) & **operator=** (const [JobDescription](#) &j) virtual ~[JobDescription](#)(void)
- std::string & **JobName** (void)
- std::list< std::string > & **Arguments** (void)
- std::list< std::string > & **Executables** (void)
- std::list< [RTE](#) > & **RuntimeEnvironments** (void)
- std::list< [RTE](#) > & **Middlewares** (void)
- std::string & **ACL** (void)
- std::string & **GMLog** (void)
- std::list< std::string > & **Loggers** (void)
- std::string & **CredentialServer** (void)
- std::string & **Stdin** (void)
- std::string & **Stdout** (void)
- std::string & **Stderr** (void)
- std::string & **Queue** (void)
- std::list< [Notification](#) > & **Notifications** (void)
- long **LifeTime** (void)
- std::list< [InputFile](#) > & **InputData** (void)
- std::list< [OutputFile](#) > & **OutputData** (void)
- int **Memory** (void)
- long **CPUTime** (void)
- long **WallTime** (void)
- long **GridTime** (void)
- int **Count** (void)
- int **Reruns** (void)
- std::string & **ClientSoftware** (void)
- std::string & **ClientHostname** (void)
- int **Disk** (void)

Protected Attributes

- std::string **job_name**
- std::list< std::string > **arguments**
- std::list< std::string > **executables**
- std::list< [RTE](#) > **rtes**
- std::list< [RTE](#) > **middlewares**
- std::string **architecture**
- std::string **acl**
- Arc::Time **start_time**
- std::string **gmlog**
- std::list< std::string > **loggers**
- std::string **credentialserver**
- std::string **cluster**

- `std::string queue`
- `std::string sstdin`
- `std::string sstdout`
- `std::string sstderr`
- `std::list< Notification > notifications`
- `long lifetime`
- `std::list< InputFile > inputdata`
- `std::list< OutputFile > outputdata`
- `int memory`
- `int disk`
- `long cpu_time`
- `long wall_time`
- `long grid_time`
- `int count`
- `int reruns`
- `std::string client_software`
- `std::string client_hostname`

Classes

- class [InputFile](#)
- class [Notification](#)
- class [OutputFile](#)

4.9.1 Detailed Description

Internal representation of [Job](#) described by JSDL

4.9.2 Member Function Documentation

4.9.2.1 `std::string& ARex2::JobDescription::JobName (void)` [inline]

Interface methods to access stored values.

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

- `job_descr.h`

4.10 ARex2::JobDescription::InputFile Class Reference

```
#include <job_descr.h>
```

Public Member Functions

- **InputFile** (const std::string &name, const std::string &source)

Public Attributes

- std::string **name**
- std::string **parameters**
- Arc::URL **source**

4.10.1 Detailed Description

Class represents the one of the input file of the job

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

- job_descr.h

4.11 ARex2::JobDescription::Notification Class Reference

```
#include <job_descr.h>
```

Public Member Functions

- **Notification** (const std::string &flags, const std::string &email)

Public Attributes

- std::string **flags**
- std::string **email**

4.11.1 Detailed Description

Class represents notification request

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

- job_descr.h

4.12 ARex2::JobDescription::OutputFile Class Reference

```
#include <job_descr.h>
```

Public Member Functions

- **OutputFile** (const std::string &name, const std::string &destination)

Public Attributes

- std::string **name**
- Arc::URL **destination**

4.12.1 Detailed Description

Class represents the one of the output file of the job

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

- job_descr.h

4.13 JobLog Class Reference

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

Public Member Functions

- **JobLog** (const char *fname)
- void **SetOutput** (const char *fname)
- bool **start_info** (JobDescription &job, const JobUser &user)
- bool **finish_info** (JobDescription &job, const JobUser &user)
- bool **is_reporting** (void)
- bool **RunReporter** (JobUsers &users)
- bool **SetReporter** (const char *destination)
- void **SetExpiration** (time_t period=0)
- bool **make_file** (JobDescription &job, JobUser &user)

Static Public Member Functions

- static bool **read_info** (std::fstream &i, bool &processed, bool &jobstart, struct tm &t, JobId &jobid, JobLocalDescription &job_desc, std::string &failure)

4.13.1 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

4.14 ARex2::JobLRMSInfo Class Reference

```
#include <lrms.h>
```

Protected Attributes

- std::string **id_**
- time_t **keep_finished**
- time_t **keep_deleted**

4.14.1 Detailed Description

Class represents the information about job in LRMS

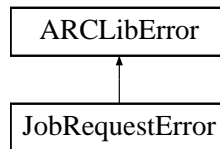
The documentation for this class was generated from the following file:

- lrms.h

4.15 JobRequestError Class Reference

```
#include <job.h>
```

Inheritance diagram for JobRequestError::



Public Member Functions

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

4.15.1 Detailed Description

Exception class thrown in case of errors with the JobRequest class.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 JobRequestError::JobRequestError (std::string *message*) [inline]

Standard exception class constructor.

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

- a-rex/grid-manager/jobdesc/job.h

4.16 JobRequestJSDL Class Reference

```
#include <job_jsdl.h>
```

Public Member Functions

- **JobRequestJSDL** (const JobRequest &j) throw (JobRequestError)
- **JobRequestJSDL** (const char *s) throw (JobRequestError)
- **JobRequestJSDL** (const std::string &s) throw (JobRequestError)
- **JobRequestJSDL** (std::istream &i) throw (JobRequestError)
- virtual JobRequest & **operator=** (const JobRequest &j) throw (JobRequestError)

Protected Member Functions

- bool **set** (std::istream &s) throw (JobRequestError)
- bool **set** (Arc::XMLNode jsdl_description_) throw (JobRequestError)
- double **get_limit** (Arc::XMLNode range)
- virtual bool **print** (std::string &s) throw (JobRequestError)

4.16.1 Detailed Description

Class to represent the request for computational job.

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

- job_jsdl.h

4.17 JobRequestXRSL Class Reference

```
#include <job_xrsl.h>
```

Public Types

- **UserFriendly**
- **NoUnits**
- enum **Type** { **UserFriendly**, **NoUnits** }

Public Member Functions

- **JobRequestXRSL** (const JobRequest &j, Type type=UserFriendly) throw (JobRequestError)
- **JobRequestXRSL** (const char *s, Type type=UserFriendly) throw (JobRequestError)
- **JobRequestXRSL** (const std::string &s, Type type=UserFriendly) throw (JobRequestError)
- **JobRequestXRSL** (std::istream &i, Type type=UserFriendly) throw (JobRequestError)
- virtual JobRequest & **operator=** (const JobRequest &j) throw (JobRequestError)

Protected Member Functions

- bool **set** (const char *s) throw (JobRequestError)
- bool **set** ([Xrsl](#) &xrsl) throw (JobRequestError)
- bool **set_xrsl** ([Xrsl](#) &xrsl) throw (JobRequestError)
- virtual bool **print** (std::string &s) throw (JobRequestError)

4.17.1 Detailed Description

Class to represent the request for computational job.

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

- `job_xrsl.h`

4.18 ARex2::JobState Class Reference

```
#include <job_state.h>
```

Public Member Functions

- **JobState** (job_state_t s)
- const std::string & **GetFailure** (void)

Protected Attributes

- job_state_t **state**
- std::string **failure_reason**

4.18.1 Detailed Description

Represents the state of job. It includes error messages as well

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

- job_state.h

4.19 ARex2::JobUser Class Reference

```
#include <job_user.h>
```

Public Member Functions

- `JobsList * get_jobs () const`
- `void operator= (JobsList *jobs_list)`
- `JobUser (const std::string &unix_name, RunPlugin *cred_plugin=NULL)`
- `JobUser (uid_t uid, RunPlugin *cred_plugin=NULL)`
- `JobUser (const JobUser &user)`
- `void SetKeepFinished (time_t ttl)`
- `void SetKeepDeleted (time_t ttr)`
- `void SetReruns (int n)`
- `void SetDiskSpace (unsigned long long int n)`
- `void SetStrictSession (bool v)`
- `void SetShareLevel (jobinfo_share_t s)`
- `bool CreateDirectories (void)`
- `time_t KeepFinished (void) const`
- `time_t KeepDeleted (void) const`
- `bool StrictSession (void) const`
- `jobinfo_share_t ShareLevel (void) const`
- `int Reruns (void) const`
- `unsigned long long int DiskSpace (void)`
- `bool SwitchUser (bool su=true) const`

4.19.1 Detailed Description

[Job](#) run under the privileges of one of the system user. This class collects information related to this user

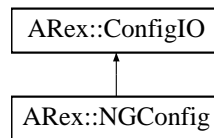
The documentation for this class was generated from the following file:

- `job_user.h`

4.20 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)

4.20.1 Detailed Description

Configuration class used for reading configuration files ARC-style.

4.20.2 Member Function Documentation

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

Read old arc.conf style configuration.

Implements [ARex::ConfigIO](#).

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

Write configuration to named file.

Implements [ARex::ConfigIO](#).

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

- ngconfig.h

4.21 HTTPD::PayloadFile Class Reference

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

Public Member Functions

- [PayloadFile](#) (const char *filename)
- [PayloadFile](#) (const char *filename, int size)
- virtual char **operator[]** (int pos) const
- virtual char * **Content** (int pos=-1)
- virtual int **Size** (void) const
- virtual char * **Insert** (int pos=0, int size=0)
- virtual char * **Insert** (const char *s, int pos=0, int size=0)
- virtual char * **Buffer** (unsigned int num)
- virtual int **BufferSize** (unsigned int num) const
- virtual int **BufferPos** (unsigned int num) const
- virtual bool **Truncate** (unsigned int size)
- **operator bool** (void)
- bool **operator!** (void)

Protected Attributes

- int **handle_**
- char * **addr_**
- size_t **size_**

4.21.1 Detailed Description

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

4.21.2 Constructor & Destructor Documentation

4.21.2.1 HTTPD::PayloadFile::PayloadFile (const char * *filename*)

Creates object associated with file for reading from it

4.21.2.2 HTTPD::PayloadFile::PayloadFile (const char * *filename*, int *size*)

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:

- httpd/PayloadFile.h

4.22 ARex::PayloadFile Class Reference

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

Public Member Functions

- [PayloadFile](#) (const char *filename, size_t start=0, size_t end=(size_t)(-1))
- [PayloadFile](#) (const char *filename, int size)
- virtual char **operator[]** (int pos) const
- virtual char * **Content** (int pos=-1)
- virtual int **Size** (void) const
- virtual char * **Insert** (int pos=0, int size=0)
- virtual char * **Insert** (const char *s, int pos=0, int size=0)
- virtual char * **Buffer** (unsigned int num)
- virtual int **BufferSize** (unsigned int num) const
- virtual int **BufferPos** (unsigned int num) const
- virtual bool **Truncate** (unsigned int size)
- **operator bool** (void)
- bool **operator!** (void)

Protected Attributes

- int **handle_**
- char * **addr_**
- size_t **size_**
- size_t **start_**
- size_t **end_**

4.22.1 Detailed Description

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

4.22.2 Constructor & Destructor Documentation

4.22.2.1 ARex::PayloadFile::PayloadFile (const char * *filename*, size_t *start* = 0, size_t *end* = (size_t)(-1))

Creates object associated with file for reading from it

4.22.2.2 ARex::PayloadFile::PayloadFile (const char * *filename*, int *size*)

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

4.23 RTE Class Reference

```
#include <rte.h>
```

Public Member Functions

- [RTE](#) (const std::string &re)
- [~RTE](#) ()
- std::string [str](#) () const
- std::string [Name](#) () const
- std::string [Version](#) () const
- bool [operator==](#) (const [RTE](#) &other) const
- bool [operator!=](#) (const [RTE](#) &other) const
- bool [operator>](#) (const [RTE](#) &other) const
- bool [operator<](#) (const [RTE](#) &other) const
- bool [operator>=](#) (const [RTE](#) &other) const
- bool [operator<=](#) (const [RTE](#) &other) const

4.23.1 Detailed Description

[RTE](#) class. It represents a runtime environment, and provides functionality for getting information about them.

4.23.2 Constructor & Destructor Documentation

4.23.2.1 RTE::RTE (const std::string &re)

Constructs a new runtime environemt. String should in general be of the type: STRING-VERSION. Where version consists of numbers with . between them.

4.23.2.2 RTE::~~RTE ()

Destructor. Not that much to say.

4.23.3 Member Function Documentation

4.23.3.1 std::string RTE::Name () const

Returns the name of the runtime environment.

4.23.3.2 bool RTE::operator!= (const [RTE](#) &other) const

Inequility operator. Return the opsite of ==

4.23.3.3 bool RTE::operator< (const [RTE](#) &other) const

Less than operator. Returns false if the other is equal, otherwise it returns the opposite of >

4.23.3.4 bool RTE::operator<= (const RTE & other) const

Less than or equal operator. Returns the oppsite of >

4.23.3.5 bool RTE::operator== (const RTE & other) const

Equiliaty operator. Returns true if the runtime environments have the string representation.

4.23.3.6 bool RTE::operator> (const RTE & other) const

Greater than operator. Returns true if the compared runtime environment is greater than the current.

4.23.3.7 bool RTE::operator>= (const RTE & other) const

Greater or equal operator. Returns the opposite of <

4.23.3.8 std::string RTE::str () const

Returns a string representation of the runtime environment. This is usually the same as given in the constructor.

4.23.3.9 std::string RTE::Version () const

Returns the version of the runtime environment.

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

- rte.h

4.24 RuntimeEnvironment Class Reference

```
#include <runtimeenvironment.h>
```

Public Member Functions

- [RuntimeEnvironment](#) (const std::string &re)
- [~RuntimeEnvironment](#) ()
- std::string [str](#) () const
- std::string [Name](#) () const
- std::string [Version](#) () const
- bool [operator==](#) (const [RuntimeEnvironment](#) &other) const
- bool [operator!=](#) (const [RuntimeEnvironment](#) &other) const
- bool [operator>](#) (const [RuntimeEnvironment](#) &other) const
- bool [operator<](#) (const [RuntimeEnvironment](#) &other) const
- bool [operator>=](#) (const [RuntimeEnvironment](#) &other) const
- bool [operator<=](#) (const [RuntimeEnvironment](#) &other) const

4.24.1 Detailed Description

[RuntimeEnvironment](#) class. It represents a runtime environment, and provides functionality for getting information about them.

4.24.2 Constructor & Destructor Documentation

4.24.2.1 RuntimeEnvironment::RuntimeEnvironment (const std::string & re)

Constructs a new runtime environemt. String should in general be of the type: STRING-VERSION. Where version consists of numbers with . between them.

4.24.2.2 RuntimeEnvironment::~~RuntimeEnvironment ()

Destructor. Not that much to say.

4.24.3 Member Function Documentation

4.24.3.1 std::string RuntimeEnvironment::Name () const

Returns the name of the runtime environment.

4.24.3.2 bool RuntimeEnvironment::operator!= (const [RuntimeEnvironment](#) & other) const

Inequality operator. Return the opsite of ==

4.24.3.3 bool RuntimeEnvironment::operator< (const [RuntimeEnvironment](#) & other) const

Less than operator. Returns false if the other is equal, otherwise it returns the opposite of >

4.24.3.4 `bool RuntimeEnvironment::operator<= (const RuntimeEnvironment & other) const`

Less than or equal operator. Returns the oppsite of >

4.24.3.5 `bool RuntimeEnvironment::operator== (const RuntimeEnvironment & other) const`

Equiliaty operator. Returns true if the runtime environments have the string representation.

4.24.3.6 `bool RuntimeEnvironment::operator> (const RuntimeEnvironment & other) const`

Greater than operator. Returns true if the compared runtime environment is greater than the current.

4.24.3.7 `bool RuntimeEnvironment::operator>= (const RuntimeEnvironment & other) const`

Greater or equal operator. Returns the opposite of <

4.24.3.8 `std::string RuntimeEnvironment::str () const`

Returns a string representation of the runtime environment. This is usually the same as given in the constructor.

4.24.3.9 `std::string RuntimeEnvironment::Version () const`

Returns the version of the runtime environment.

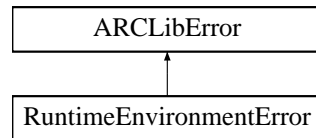
The documentation for this class was generated from the following file:

- runtimeenvironment.h

4.25 `RuntimeEnvironmentError` Class Reference

```
#include <runtimeenvironment.h>
```

Inheritance diagram for `RuntimeEnvironmentError`:



Public Member Functions

- [`RuntimeEnvironmentError`](#) (`std::string message`)

4.25.1 Detailed Description

[`RuntimeEnvironment`](#) exceptions. Gets thrown when an error occurs regarding a runtime environment.

4.25.2 Constructor & Destructor Documentation

4.25.2.1 `RuntimeEnvironmentError::RuntimeEnvironmentError` (`std::string message`) [inline]

Standard exception class constructor.

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

- `runtimeenvironment.h`

4.26 ArcSec::Service_AA Class Reference

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

Public Member Functions

- **Service_AA** (Arc::Config *cfg)
- virtual Arc::MCC_Status **process** (Arc::Message &inmsg, Arc::Message &outmsg)

Protected Member Functions

- Arc::MCC_Status **make_soap_fault** (Arc::Message &outmsg)

Protected Attributes

- Arc::NS **ns_**
- Arc::Logger **logger_**

4.26.1 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

4.27 ArcSec::Service_PDP Class Reference

```
#include <pdpservice.h>
```

Public Member Functions

- **Service_PDP** (Arc::Config *cfg)
- virtual Arc::MCC_Status **process** (Arc::Message &inmsg, Arc::Message &outmsg)

Protected Member Functions

- Arc::MCC_Status **make_soap_fault** (Arc::Message &outmsg)

Protected Attributes

- Arc::NS **ns_**
- Arc::Logger **logger_**

4.27.1 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:

- pdpservice.h

4.28 Time Class Reference

```
#include <datetime.h>
```

Public Member Functions

- [Time](#) ()
- [Time](#) (const time_t &)
- [Time](#) (const std::string &)
- [Time](#) & [operator=](#) (const time_t &)
- void [SetTime](#) (const time_t &)
- time_t [GetTime](#) () const
- [operator](#) std::string () const
- std::string [str](#) (const TimeFormat &=time_format) const
- bool [operator<](#) (const [Time](#) &) const
- bool [operator>](#) (const [Time](#) &) const
- bool [operator<=](#) (const [Time](#) &) const
- bool [operator>=](#) (const [Time](#) &) const
- bool [operator==](#) (const [Time](#) &) const
- bool [operator!=](#) (const [Time](#) &) const

Static Public Member Functions

- static void [SetFormat](#) (const TimeFormat &)
- static TimeFormat [GetFormat](#) ()

4.28.1 Detailed Description

A class for storing and manipulating times.

4.28.2 Constructor & Destructor Documentation

4.28.2.1 Time::Time ()

Default constructor. The time is put equal the current time.

4.28.2.2 Time::Time (const time_t &)

Constructor that takes a time_t variable and stores it.

4.28.2.3 Time::Time (const std::string &)

Constructor that tries to convert a string into a time_t.

4.28.3 Member Function Documentation

4.28.3.1 static TimeFormat Time::GetFormat () [static]

Gets the default format for time strings.

4.28.3.2 time_t Time::GetTime () const

gets the time

4.28.3.3 Time::operator std::string () const

Returns a string representation of the time, using the default format.

4.28.3.4 bool Time::operator!= (const Time &) const

Comparing two Time objects.

4.28.3.5 bool Time::operator< (const Time &) const

Comparing two Time objects.

4.28.3.6 bool Time::operator<= (const Time &) const

Comparing two Time objects.

4.28.3.7 Time& Time::operator= (const time_t &)

Assignment operator from a time_t.

4.28.3.8 bool Time::operator== (const Time &) const

Comparing two Time objects.

4.28.3.9 bool Time::operator> (const Time &) const

Comparing two Time objects.

4.28.3.10 bool Time::operator>= (const Time &) const

Comparing two Time objects.

4.28.3.11 static void Time::SetFormat (const TimeFormat &) [static]

Sets the default format for time strings.

4.28.3.12 void Time::SetTime (const time_t &)

sets the time

4.28.3.13 std::string Time::str (const TimeFormat & = time_format) const

Returns a string representation of the time, using the specified format.

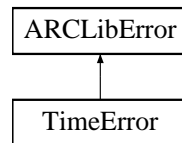
The documentation for this class was generated from the following file:

- datetime.h

4.29 TimeError Class Reference

```
#include <datetime.h>
```

Inheritance diagram for TimeError::



Public Member Functions

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

4.29.1 Detailed Description

Class to represent errors thrown by the [Time](#) class.

4.29.2 Constructor & Destructor Documentation

4.29.2.1 TimeError::TimeError (std::string *message*) [inline]

Standard exception class constructor.

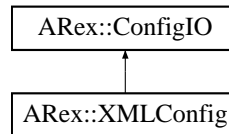
The documentation for this class was generated from the following file:

- datetime.h

4.30 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)

4.30.1 Detailed Description

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

4.30.2 Member Function Documentation

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

Read configuration.

Implements [ARex::ConfigIO](#).

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

Write configuration.

Implements [ARex::ConfigIO](#).

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

- xmlconfig.h

4.31 Xrsl Class Reference

```
#include <xrsl.h>
```

Public Member Functions

- [Xrsl](#) (const std::string &xrsl_string) throw (XrslError)
- [Xrsl](#) (xrsl_operator=operator_and)
- [Xrsl](#) (globus_rsl_t *)
- [Xrsl](#) (const [Xrsl](#) &other_xrsl)
- [Xrsl](#) & [operator=](#) (const [Xrsl](#) &other_xrsl)
- [~Xrsl](#) ()
- void [Print](#) () const
- const std::string [str](#) () const throw (XrslError)
- std::list< [Xrsl](#) > [SplitMulti](#) ()
- std::list< [Xrsl](#) > [SplitOrRelation](#) () throw (XrslError)
- void [AddRelation](#) (const [XrslRelation](#) &relation, bool force=true) throw (XrslError)
- void [AddSimpleRelation](#) (const std::string &attr, xrsl_operator op, const std::string &val, bool force=true) throw (XrslError)
- void [AddXrsl](#) ([Xrsl](#) &axrsl) throw (XrslError)
- [XrslRelation](#) [GetRelation](#) (const std::string &attr) throw (XrslError)
- std::list< [XrslRelation](#) > [GetAllRelations](#) (const std::string &attr)
- bool [IsRelation](#) (const std::string &)
- void [RemoveRelation](#) (const std::string &attr) throw (XrslError)
- void [Validate](#) (const std::list< [XrslValidationData](#) > &valid_attributes, bool allow_unknown=false) throw (XrslError)
- void [Eval](#) ()

4.31.1 Detailed Description

Class used to simplify manipulation of xRSL job descriptions.

4.31.2 Constructor & Destructor Documentation

4.31.2.1 Xrsl::Xrsl (const std::string & *xrsl_string*) throw ([XrslError](#))

Constructs a [Xrsl](#) object from a string representation.

4.31.2.2 Xrsl::Xrsl (xrsl_operator = operator_and)

Constructs empty [Xrsl](#) object.

4.31.2.3 Xrsl::Xrsl (globus_rsl_t *)

Construct [Xrsl](#) object from globus_rsl_t* .

4.31.2.4 `Xrsl::Xrsl (const Xrsl & other_xrsl)`

Copy constructor.

4.31.2.5 `Xrsl::~~Xrsl ()`

Destructor.

4.31.3 Member Function Documentation**4.31.3.1** `void Xrsl::AddRelation (const XrslRelation & relation, bool force = true) throw (XrslError)`

Adds a new relation. Throws exception if relation already exists in the xrsl and force is not true.

4.31.3.2 `void Xrsl::AddSimpleRelation (const std::string & attr, xrsl_operator op, const std::string & val, bool force = true) throw (XrslError)`

Adds simple relation specified by attribute, xrsl-operator and value. Throws exception if relation already exists in the xrsl and force is not true.

4.31.3.3 `void Xrsl::AddXrsl (Xrsl & axrsl) throw (XrslError)`

Adds a sub-Xrsl to the [Xrsl](#).

4.31.3.4 `void Xrsl::Eval ()`

Performs RSL alias substitution etc.

4.31.3.5 `std::list<XrslRelation> Xrsl::GetAllRelations (const std::string & attr)`

Get all [XrslRelation](#)'s in the xrsl with attribute equal to parameter attr.

4.31.3.6 `XrslRelation Xrsl::GetRelation (const std::string & attr) throw (XrslError)`

Gets the first [XrslRelation](#) corresponding to the attribute.

4.31.3.7 `bool Xrsl::IsRelation (const std::string &)`

Does the relation with this attribute exist?

4.31.3.8 `Xrsl & Xrsl::operator= (const Xrsl & other_xrsl)`

Copy-assignment constructor.

4.31.3.9 void Xrsl::Print () const

Print detailed information about each relation.

4.31.3.10 void Xrsl::RemoveRelation (const std::string & attr) throw (XrslError)

Removes a relation. Throws an exception if the relation does not exist in the xrsl. The relation may be of any type.

4.31.3.11 std::list<Xrsl> Xrsl::SplitMulti ()

If the [Xrsl](#) start with a +, split the [Xrsl](#) into multiple Xrsls.

4.31.3.12 std::list<Xrsl> Xrsl::SplitOrRelation () throw (XrslError)

Splits an [Xrsl](#) containing or-operators into separate [Xrsl](#)'s. Example:
&(executable=/bin/echo)(|(cluster=c1)(cluster=c2)) split into &(executable=/bin/echo)(cluster=c1)
and &(executable=/bin/echo)(cluster=c2).

4.31.3.13 const std::string Xrsl::str () const throw (XrslError)

Converts the [Xrsl](#) object to std::string representation.

4.31.3.14 void Xrsl::Validate (const std::list< XrslValidationData > & valid_attributes, bool allow_unknown = false) throw (XrslError)

Ensures that the xrsl only contains valid attributes. Throws exception if some attribute has invalid format, or that some mandatory attribute is missing.

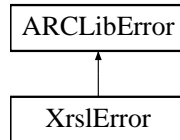
The documentation for this class was generated from the following file:

- xrsl.h

4.32 XrslError Class Reference

```
#include <xrsl.h>
```

Inheritance diagram for XrslError::



Public Member Functions

- [XrslError](#) (std::string message_arg)

4.32.1 Detailed Description

Class represents exceptions associated with usage of the [Xrsl](#) class.

4.32.2 Constructor & Destructor Documentation

4.32.2.1 XrslError::XrslError (std::string *message_arg*) [inline]

Standard exception constructor.

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

- xrsl.h

4.33 XrslRelation Class Reference

```
#include <xrsl.h>
```

Public Member Functions

- [XrslRelation](#) (const std::string &attribute, const xrsl_operator &oper, const std::string &value)
- [XrslRelation](#) (const std::string &attribute, const xrsl_operator &oper, const std::list< std::string > &value)
- [XrslRelation](#) (const std::string &attribute, const xrsl_operator &oper, const std::list< std::list< std::string > > &value)
- [XrslRelation](#) (globus_rsl_t *relation)
- [XrslRelation](#) (const [XrslRelation](#) &other_relation)
- [~XrslRelation](#) ()
- std::string [str](#) ()
- std::string [GetAttribute](#) () const
- xrsl_operator [GetOperator](#) () const
- std::string [GetSingleValue](#) () throw (XrslError)
- std::list< std::string > [GetListValue](#) () throw (XrslError)
- std::list< std::list< std::string > > [GetDoubleListValue](#) () throw (XrslError)
- globus_rsl_t * [GetRelation](#) () const

4.33.1 Detailed Description

[XrslRelation](#) class that describes an Xrsl-relation with an attribute, an operator and a value. Various constructors and methods for extracting the attribute and the value (single value or list) are given.

4.33.2 Constructor & Destructor Documentation

4.33.2.1 XrslRelation::XrslRelation (const std::string & *attribute*, const xrsl_operator & *oper*, const std::string & *value*)

Constructor constructing an xrsl-relation from an attribute, an operator and a value.

4.33.2.2 XrslRelation::XrslRelation (const std::string & *attribute*, const xrsl_operator & *oper*, const std::list< std::string > & *value*)

Constructor constructing an xrsl-relation from an attribute, an operator and a value-list.

4.33.2.3 XrslRelation::XrslRelation (const std::string & *attribute*, const xrsl_operator & *oper*, const std::list< std::list< std::string > > & *value*)

Constructor constructing an xrsl-relation from an attribute, an operator and a double value-list.

4.33.2.4 XrslRelation::XrslRelation (globus_rsl_t * *relation*)

Constructs a relation from a globus_rsl_t*.

4.33.2.5 XrslRelation::XrslRelation (const [XrslRelation](#) & *other_relation*)

Copy-constructor.

4.33.2.6 XrslRelation::~~XrslRelation ()

Destructor.

4.33.3 Member Function Documentation**4.33.3.1 std::string XrslRelation::GetAttribute () const**

Returns the attribute of the relation.

4.33.3.2 std::list<std::list<std::string> > XrslRelation::GetDoubleListValue () throw ([XrslError](#))

If the value of the relation is a double list value, return it.

4.33.3.3 std::list<std::string> XrslRelation::GetListValue () throw ([XrslError](#))

If the value of the relation is a list value, return it.

4.33.3.4 xrsl_operator XrslRelation::GetOperator () const

Returns the xrsl_operator of the attribute.

4.33.3.5 globus_rsl_t* XrslRelation::GetRelation () const

Returns relation.

4.33.3.6 std::string XrslRelation::GetSingleValue () throw ([XrslError](#))

If the value of the representation is a single string value, return it.

4.33.3.7 std::string XrslRelation::str ()

Returns a string representation of the relation.

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

- xrsl.h

4.34 XrslValidationData Class Reference

```
#include <xrsl.h>
```

Public Member Functions

- **XrslValidationData** (const std::string &*attribute_name*, relation_type *rel_type*, validation_type *val_type*, bool *unique*=true, int *list_length*=0)

Public Attributes

- std::string *attribute_name*
- relation_type *rel_type*
- bool *unique*
- int *list_length*
- validation_type *val_type*

4.34.1 Detailed Description

Class for simplifying **Xrsl** validation. One object of this class represents a valid attribute in the **xrsl**.

4.34.2 Constructor & Destructor Documentation

- 4.34.2.1** **XrslValidationData::XrslValidationData** (const std::string & *attribute_name*, relation_type *rel_type*, validation_type *val_type*, bool *unique* = true, int *list_length* = 0)

Constructor.

4.34.3 Member Data Documentation

- 4.34.3.1** std::string **XrslValidationData::attribute_name**

Name of attribute.

- 4.34.3.2** int **XrslValidationData::list_length**

Length of each list in case attribute is a list of values.

- 4.34.3.3** relation_type **XrslValidationData::rel_type**

Type the attribute must have.

- 4.34.3.4** bool **XrslValidationData::unique**

Must this attribute be unique?

4.34.3.5 validation_type [XrslValidationData::val_type](#)

Must the [Xrsl](#) must contain this attribute to be valid?

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

- xrsl.h

Chapter 5

Hosting Environment (Daemon) Services File Documentation

5.1 configcore.h File Reference

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

Namespaces

- namespace **ARex**

Classes

- 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)

Variables

- Arc::Logger **ARex::ConfigLogger**

5.1.1 Detailed Description

This file describes the core configuration

Index

- ~ARCLibError
 - ARCLibError, [9](#)
- ~Job
 - Job, [16](#)
- ~RTE
 - RTE, [35](#)
- ~RuntimeEnvironment
 - RuntimeEnvironment, [37](#)
- ~Xrsl
 - Xrsl, [48](#)
- ~XrslRelation
 - XrslRelation, [52](#)
- AddRelation
 - Xrsl, [48](#)
- AddSimpleRelation
 - Xrsl, [48](#)
- AddXrsl
 - Xrsl, [48](#)
- ARCLibError, [9](#)
 - ARCLibError, [9](#)
- ARCLibError
 - ~ARCLibError, [9](#)
 - ARCLibError, [9](#)
 - what, [10](#)
- ArcSec::Service_AA, [40](#)
- ArcSec::Service_PDP, [41](#)
- ARex2::JobControl, [18](#)
- ARex2::JobDataCache, [19](#)
- ARex2::JobDescription, [20](#)
- ARex2::JobDescription
 - JobName, [21](#)
- ARex2::JobDescription::InputFile, [22](#)
- ARex2::JobDescription::Notification, [23](#)
- ARex2::JobDescription::OutputFile, [24](#)
- ARex2::JobLRMSInfo, [26](#)
- ARex2::JobState, [30](#)
- ARex2::JobUser, [31](#)
- ARex::ARexJob, [11](#)
- ARex::ARexJob
 - ARexJob, [11](#)
 - Cancel, [11](#)
 - Clean, [11](#)
 - Failed, [12](#)
 - Failure, [12](#)
 - GetDescription, [12](#)
 - ID, [12](#)
 - Resume, [12](#)
 - SessionDir, [12](#)
 - State, [12](#)
 - TotalJobs, [12](#)
- ARex::Config, [13](#)
 - ConfValue, [13](#)
 - FirstConfValue, [13](#)
 - GetConfigs, [13](#)
- ARex::ConfigError, [14](#)
- ARex::ConfigError
 - ConfigError, [14](#)
- ARex::ConfigIO, [15](#)
 - Read, [15](#)
 - Write, [15](#)
- ARex::NGConfig, [32](#)
 - Read, [32](#)
 - Write, [32](#)
- ARex::PayloadFile, [34](#)
- ARex::PayloadFile
 - PayloadFile, [34](#)
- ARex::XMLConfig, [46](#)
 - Read, [46](#)
 - Write, [46](#)
- ARexJob
 - ARex::ARexJob, [11](#)
- attribute_name
 - XrslValidationData, [53](#)
- Cancel
 - ARex::ARexJob, [11](#)
 - Job, [16](#)
- Clean
 - ARex::ARexJob, [11](#)
- configcore.h, [55](#)
- ConfigError
 - ARex::ConfigError, [14](#)
- ConfValue
 - ARex::Config, [13](#)
- Eval
 - Xrsl, [48](#)
- Failed

- ARex::ARexJob, 12
- Failure
 - ARex::ARexJob, 12
- FirstConfValue
 - ARex::Config, 13
- GetAllRelations
 - Xrsl, 48
- GetAttribute
 - XrslRelation, 52
- GetConfigs
 - ARex::Config, 13
- GetDescription
 - ARex::ARexJob, 12
- GetDoubleListValue
 - XrslRelation, 52
- GetFormat
 - Time, 43
- GetListValue
 - XrslRelation, 52
- GetOperator
 - XrslRelation, 52
- GetRelation
 - Xrsl, 48
 - XrslRelation, 52
- GetSessionDir
 - Job, 16
- GetSingleValue
 - XrslRelation, 52
- GetState
 - Job, 16
- GetTime
 - Time, 43
- HTTPD::PayloadFile, 33
- HTTPD::PayloadFile
 - PayloadFile, 33
- ID
 - ARex::ARexJob, 12
- IsRelation
 - Xrsl, 48
- Job, 16
 - ~Job, 16
 - Cancel, 16
 - GetSessionDir, 16
 - GetState, 16
 - Job, 16
 - operator bool, 17
 - Resume, 17
 - Start, 17
- JobLog, 25
- JobName
 - ARex2::JobDescription, 21
- JobRequestError, 27
 - JobRequestError, 27
- JobRequestError
 - JobRequestError, 27
- JobRequestJSDL, 28
- JobRequestXRSL, 29
- list_length
 - XrslValidationData, 53
- Name
 - RTE, 35
 - RuntimeEnvironment, 37
- operator bool
 - Job, 17
- operator std::string
 - Time, 43
- operator!=
 - RTE, 35
 - RuntimeEnvironment, 37
 - Time, 43
- operator<
 - RTE, 35
 - RuntimeEnvironment, 37
 - Time, 43
- operator<=
 - RTE, 35
 - RuntimeEnvironment, 37
 - Time, 43
- operator=
 - Time, 43
 - Xrsl, 48
- operator==
 - RTE, 36
 - RuntimeEnvironment, 38
 - Time, 43
- operator>
 - RTE, 36
 - RuntimeEnvironment, 38
 - Time, 43
- operator>=
 - RTE, 36
 - RuntimeEnvironment, 38
 - Time, 43
- PayloadFile
 - ARex::PayloadFile, 34
 - HTTPD::PayloadFile, 33
- Print
 - Xrsl, 48
- Read
 - ARex::ConfigIO, 15

- ARex::NGConfig, 32
- ARex::XMLConfig, 46
- rel_type
 - XrslValidationData, 53
- RemoveRelation
 - Xrsl, 49
- Resume
 - ARex::ARexJob, 12
 - Job, 17
- RTE, 35
 - ~RTE, 35
 - Name, 35
 - operator!=, 35
 - operator<, 35
 - operator<=, 35
 - operator==, 36
 - operator>, 36
 - operator>=, 36
 - RTE, 35
 - str, 36
 - Version, 36
- RuntimeEnvironment, 37
 - RuntimeEnvironment, 37
- RuntimeEnvironment
 - ~RuntimeEnvironment, 37
 - Name, 37
 - operator!=, 37
 - operator<, 37
 - operator<=, 37
 - operator==, 38
 - operator>, 38
 - operator>=, 38
 - RuntimeEnvironment, 37
 - str, 38
 - Version, 38
- RuntimeEnvironmentError, 39
 - RuntimeEnvironmentError, 39
- RuntimeEnvironmentError
 - RuntimeEnvironmentError, 39
- SessionDir
 - ARex::ARexJob, 12
- SetFormat
 - Time, 43
- SetTime
 - Time, 43
- SplitMulti
 - Xrsl, 49
- SplitOrRelation
 - Xrsl, 49
- Start
 - Job, 17
- State
 - ARex::ARexJob, 12
- str
 - RTE, 36
 - RuntimeEnvironment, 38
 - Time, 44
 - Xrsl, 49
 - XrslRelation, 52
- Time, 42
 - GetFormat, 43
 - GetTime, 43
 - operator std::string, 43
 - operator!=, 43
 - operator<, 43
 - operator<=, 43
 - operator==, 43
 - operator>, 43
 - operator>=, 43
 - SetFormat, 43
 - SetTime, 43
 - str, 44
 - Time, 42
- TimeError, 45
 - TimeError, 45
- TimeError
 - TimeError, 45
- TotalJobs
 - ARex::ARexJob, 12
- unique
 - XrslValidationData, 53
- val_type
 - XrslValidationData, 53
- Validate
 - Xrsl, 49
- Version
 - RTE, 36
 - RuntimeEnvironment, 38
- what
 - ARCLibError, 10
- Write
 - ARex::ConfigIO, 15
 - ARex::NGConfig, 32
 - ARex::XMLConfig, 46
- Xrsl, 47
 - ~Xrsl, 48
 - AddRelation, 48
 - AddSimpleRelation, 48
 - AddXrsl, 48
 - Eval, 48
 - GetAllRelations, 48
 - GetRelation, 48

- IsRelation, [48](#)
- operator=, [48](#)
- Print, [48](#)
- RemoveRelation, [49](#)
- SplitMulti, [49](#)
- SplitOrRelation, [49](#)
- str, [49](#)
- Validate, [49](#)
- Xrsl, [47](#)
- XrslError, [50](#)
 - XrslError, [50](#)
- XrslError
 - XrslError, [50](#)
- XrslRelation, [51](#)
 - XrslRelation, [51](#)
- XrslRelation
 - ~XrslRelation, [52](#)
 - GetAttribute, [52](#)
 - GetDoubleListValue, [52](#)
 - GetListValue, [52](#)
 - GetOperator, [52](#)
 - GetRelation, [52](#)
 - GetSingleValue, [52](#)
 - str, [52](#)
 - XrslRelation, [51](#)
- XrslValidationData, [53](#)
 - XrslValidationData, [53](#)
- XrslValidationData
 - attribute_name, [53](#)
 - list_length, [53](#)
 - rel_type, [53](#)
 - unique, [53](#)
 - val_type, [53](#)
 - XrslValidationData, [53](#)