Hosting Environment (Daemon) Services

Generated by Doxygen 1.7.1

Thu Dec 9 2010 14:24:24

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Data Structure Index **3**

2.1 Class Hierarchy 3

3 Data Structure Index **7**

3.1 Data Structures 7

4 File Index **11**

4.1 File List 11

5 Namespace Documentation **15**

5.1 DREService Namespace Reference 15

5.1.1 Detailed Description 15

6 Data Structure Documentation **17**

6.1 ARex2::ARex2Service Class Reference 17

6.2 ARex::ARexGMConfig Class Reference 17

6.3 ARex::ARexJob Class Reference 17

6.3.1 Detailed Description 18

6.3.2 Constructor & Destructor Documentation 18

6.3.2.1 ARexJob 18

6.3.2.2 ARexJob 18

6.3.3 Member Function Documentation 18

6.3.3.1 Cancel 18

6.3.3.2 ChooseSessionDir 18

6.3.3.3 Clean 18

6.3.3.4 CreateFile 19

6.3.3.5 Failed 19

6.3.3.6 Failure . . . 19
6.3.3.7 GetDescription . . . 19
6.3.3.8 ID . . . 19
6.3.3.9 Jobs . . . 19
6.3.3.10 LogDir . . . 19
6.3.3.11 LogFiles . . . 19
6.3.3.12 OpenDir . . . 19
6.3.3.13 OpenFile . . . 19
6.3.3.14 OpenLogFile . . . 19
6.3.3.15 Resume . . . 20
6.3.3.16 SessionDir . . . 20
6.3.3.17 State . . . 20
6.3.3.18 State . . . 20
6.3.3.19 TotalJobs . . . 20
6.3.3.20 UpdateCredentials . . . 20
6.4 ARex::ARexService Class Reference . . . 20
6.5 AuthEvaluator Class Reference . . . 20
6.6 AuthUser Class Reference . . . 20
6.7 AuthVO Class Reference . . . 21
6.8 CacheConfig Class Reference . . . 21
6.8.1 Detailed Description . . . 21
6.8.2 Constructor & Destructor Documentation . . . 21
6.8.2.1 CacheConfig . . . 21
6.8.2.2 CacheConfig . . . 21
6.8.3 Member Function Documentation . . . 22
6.8.3.1 parseINIConf . . . 22
6.8.3.2 setCacheDirs . . . 22
6.9 CacheConfigException Class Reference . . . 22
6.9.1 Detailed Description . . . 22
6.10 Cache::CacheService Class Reference . . . 22
6.10.1 Detailed Description . . . 23
6.10.2 Constructor & Destructor Documentation . . . 23
6.10.2.1 CacheService . . . 23
6.10.2.2 ∼CacheService . . . 23
6.10.3 Member Function Documentation . . . 23
6.10.3.1 CacheCheck . . . 23

6.10.3.2 CacheLink . . . 23
6.10.3.3 operator bool . . . 23
6.10.3.4 operator! . . . 24
6.10.3.5 process . . . 24
6.10.3.6 RegistrationCollector . . . 24
6.11 ArcSec::Charon Class Reference . . . 24
6.11.1 Detailed Description . . . 24
6.12 CommFIFO Class Reference . . . 24
6.13 Arc::Compiler_AREXClient Class Reference . . . 25
6.13.1 Detailed Description . . . 25
6.13.2 Constructor & Destructor Documentation . . . 25
6.13.2.1 Compiler_AREXClient . . . 25
6.13.2.2 ∼Compiler_AREXClient . . . 26
6.13.3 Member Function Documentation . . . 26
6.13.3.1 clean . . . 26
6.13.3.2 kill . . . 26
6.13.3.3 sstat . . . 26
6.13.3.4 stat . . . 27
6.13.3.5 submit . . . 27
6.14 Arc::Compiler_AREXClientError Class Reference . . . 27
6.14.1 Detailed Description . . . 28
6.14.2 Constructor & Destructor Documentation . . . 28
6.14.2.1 Compiler_AREXClientError . . . 28
6.15 Arc::Compiler_AREXFile Class Reference . . . 28
6.16 ARex::ConfGrp Class Reference . . . 28
6.17 ARex::Config Class Reference . . . 28
6.17.1 Detailed Description . . . 28
6.17.2 Member Function Documentation . . . 29
6.17.2.1 ConfValue . . . 29
6.17.2.2 FirstConfValue . . . 29
6.17.2.3 GetConfigs . . . 29
6.18 ARex::ConfigError Class Reference . . . 29
6.18.1 Detailed Description . . . 29
6.18.2 Constructor & Destructor Documentation . . . 29
6.18.2.1 ConfigError . . . 29
6.19 ARex::ConfigIO Class Reference . . . 29

6.19.1 Detailed Description . . . 30
6.19.2 Member Function Documentation . . . 30
6.19.2.1 Read . . . 30
6.19.2.2 Write . . . 30
6.20 ConfigSections Class Reference . . . 30
6.21 gridftpd::ConfigSections Class Reference . . . 30
6.22 Paul::Configurator Class Reference . . . 31
6.23 ContinuationPlugins Class Reference . . . 31
6.24 ARex::CountedResource Class Reference . . . 31
6.25 gridftpd::Daemon Class Reference . . . 31
6.26 DirectAccess::diraccess_t Struct Reference . . . 31
6.27 DirectAccess Class Reference . . . 31
6.28 DirectFilePlugin Class Reference . . . 32
6.29 DirEntry Class Reference . . . 32
6.30 DREService::DREWebService Class Reference . . . 32
6.30.1 Constructor & Destructor Documentation . . . 32
6.30.1.1 DREWebService . . . 32
6.30.1.2 ∼DREWebService . . . 33
6.30.2 Member Function Documentation . . . 33
6.30.2.1 makeFault . . . 33
6.30.2.2 process . . . 33
6.30.3 Field Documentation . . . 33
6.30.3.1 logger . . . 33
6.30.3.2 ns_ . . . 33
6.31 Entry Class Reference . . . 34
6.32 ARex::FileChunks Class Reference . . . 34
6.32.1 Detailed Description . . . 34
6.32.2 Member Function Documentation . . . 34
6.32.2.1 Release . . . 34
6.32.2.2 Remove . . . 34
6.33 ARex::FileChunksList Class Reference . . . 34
6.33.1 Detailed Description . . . 35
6.33.2 Member Function Documentation . . . 35
6.33.2.1 Get . . . 35
6.33.2.2 GetStuck . . . 35
6.34 FileData Class Reference . . . 35

6.35 FileNode Class Reference . . . 35
6.36 FilePlugin Class Reference . . . 35
6.37 FileRoot Class Reference . . . 36
6.38 fs_usage Struct Reference . . . 36
6.39 GACLPlugin Class Reference . . . 36
6.40 gm_dirs_ Struct Reference . . . 37
6.41 GMEnvironment Class Reference . . . 37
6.41.1 Member Function Documentation . . . 37
6.41.1.1 nordugrid_config_loc . . . 37
6.41.1.2 support_mail_address . . . 37
6.42 gridftpd::GMEnvironment Class Reference . . . 37
6.42.1 Member Function Documentation . . . 37
6.42.1.1 nordugrid_config_loc . . . 37
6.42.1.2 support_mail_address . . . 38
6.43 GridFTP_Commands Class Reference . . . 38
6.44 GridFTP_Commands_timeout Class Reference . . . 38
6.45 ARex::GridManager Class Reference . . . 38
6.46 GridScheduler::GridSchedulerService Class Reference . . . 38
6.47 Hopi::Hopi Class Reference . . . 38
6.48 Paul::HTMLRequest Class Reference . . . 39
6.49 Paul::HTMLResponse Class Reference . . . 39
6.50 Identity Class Reference . . . 39
6.51 IdentityGACL Class Reference . . . 39
6.52 IdentityItemDN Class Reference . . . 40
6.53 IdentityItemVOMS Class Reference . . . 40
6.54 Index Class Reference . . . 40
6.55 ARex2::JobDescription::InputFile Class Reference . . . 40
6.55.1 Detailed Description . . . 40
6.56 Paul::InvalidMessageException Class Reference . . . 41
6.57 ISIS::ISIService Class Reference . . . 41
6.58 ISIS::ISISSecAttr Class Reference . . . 41
6.59 Identity::Item Class Reference . . . 41
6.60 ObjectAccess::Item Class Reference . . . 41
6.61 Janitor Class Reference . . . 42
6.61.1 Detailed Description . . . 42
6.61.2 Constructor & Destructor Documentation . . . 42

6.61.2.1 Janitor . . . 42
6.61.3 Member Function Documentation . . . 42
6.61.3.1 deploy . . . 42
6.61.3.2 remove . . . 42
6.61.3.3 result . . . 43
6.61.3.4 wait . . . 43
6.62 Paul::Job Class Reference . . . 43
6.63 Arc::Job Class Reference . . . 43
6.64 Job Class Reference . . . 43
6.64.1 Detailed Description . . . 43
6.64.2 Constructor & Destructor Documentation . . . 44
6.64.2.1 Job . . . 44
6.64.2.2 Job . . . 44
6.64.2.3 ∼Job . . . 44
6.64.3 Member Function Documentation . . . 44
6.64.3.1 Cancel . . . 44
6.64.3.2 GetSessionDir . . . 44
6.64.3.3 GetState . . . 44
6.64.3.4 operator bool . . . 44
6.64.3.5 Resume . . . 44
6.64.3.6 Start . . . 44
6.65 job_state_rec_t Struct Reference . . . 45
6.66 ARex2::JobControl Class Reference . . . 45
6.66.1 Detailed Description . . . 45
6.67 ARex2::JobController Class Reference . . . 45
6.68 ARex2::JobDataCache Class Reference . . . 45
6.68.1 Detailed Description . . . 45
6.69 JobDescription Class Reference . . . 45
6.70 ARex2::JobDescription Class Reference . . . 46
6.70.1 Detailed Description . . . 46
6.70.2 Member Function Documentation . . . 46
6.70.2.1 JobName . . . 46
6.71 Paul::JobList Class Reference . . . 46
6.72 JobLocalDescription Class Reference . . . 46
6.73 JobLog Class Reference . . . 46
6.73.1 Detailed Description . . . 47

6.74 ARex2::JobLRMSInfo Class Reference . . . 47
6.74.1 Detailed Description . . . 47
6.75 Arc::JobNotFoundException Class Reference . . . 47
6.76 JobPlugin Class Reference . . . 47
6.77 Paul::JobQueue Class Reference . . . 47
6.78 Arc::JobQueue Class Reference . . . 48
6.79 Arc::JobQueueIterator Class Reference . . . 48
6.80 ARex::JobRecord Class Reference . . . 48
6.81 Paul::JobRequest Class Reference . . . 48
6.82 Arc::JobRequest Class Reference . . . 48
6.83 Arc::JobSchedMetaData Class Reference . . . 48
6.84 Paul::JobSchedMetaData Class Reference . . . 48
6.85 Arc::JobSelector Class Reference . . . 49
6.86 JobsList Class Reference . . . 49
6.87 JobsListConfig Class Reference . . . 49
6.87.1 Detailed Description . . . 49
6.88 ARex2::JobState Class Reference . . . 49
6.88.1 Detailed Description . . . 49
6.89 JobUser Class Reference . . . 49
6.90 ARex2::JobUser Class Reference . . . 50
6.90.1 Detailed Description . . . 50
6.91 JobUserHelper Class Reference . . . 50
6.92 JobUsers Class Reference . . . 50
6.93 gridftpd::LdapQuery Class Reference . . . 50
6.93.1 Detailed Description . . . 51
6.93.2 Member Enumeration Documentation . . . 51
6.93.2.1 Scope . . . 51
6.93.3 Constructor & Destructor Documentation . . . 51
6.93.3.1 LdapQuery . . . 51
6.93.3.2 ∼LdapQuery . . . 51
6.93.4 Member Function Documentation . . . 51
6.93.4.1 Host . . . 51
6.93.4.2 Query . . . 51
6.93.4.3 Result . . . 51
6.94 gridftpd::LdapQueryError Class Reference . . . 51
6.94.1 Detailed Description . . . 52

6.94.2 Constructor & Destructor Documentation . . . 52
6.94.2.1 LdapQueryError . . . 52
6.95 RunPlugin::lib_plugin_t Union Reference . . . 52
6.96 gridftpd::RunPlugin::lib_plugin_t Union Reference . . . 52
6.97 ARex::LoggerClient Class Reference . . . 52
6.98 LRMSResult Class Reference . . . 52
6.99 ISIS::Neighbor_Container Class Reference . . . 53
6.100 ARex::NGConfig Class Reference . . . 53
6.100.1 Detailed Description . . . 53
6.100.2 Member Function Documentation . . . 53
6.100.2.1 Read . . . 53
6.100.2.2 Write . . . 53
6.101 ARex2::JobDescription::Notification Class Reference . . . 53
6.101.1 Detailed Description . . . 54
6.102 numvalue_for_shell Class Reference . . . 54
6.103 ObjectAccess Class Reference . . . 54
6.104 ObjectAccessGACL Class Reference . . . 54
6.105 ARex::OptimizedInformationContainer Class Reference . . . 55
6.106 ARex::Option Class Reference . . . 55
6.107 ARex2::JobDescription::OutputFile Class Reference . . . 55
6.107.1 Detailed Description . . . 55
6.108 gridftpd::ParallelLdapQueries Class Reference . . . 55
6.108.1 Detailed Description . . . 55
6.109 Paul::PaulService Class Reference . . . 55
6.110 ARex::PayloadBigFile Class Reference . . . 56
6.110.1 Constructor & Destructor Documentation . . . 56
6.110.1.1 PayloadBigFile . . . 56
6.110.1.2 ~PayloadBigFile . . . 56
6.111 Hopi::PayloadBigFile Class Reference . . . 56
6.111.1 Constructor & Destructor Documentation . . . 56
6.111.1.1 PayloadBigFile . . . 56
6.111.1.2 ~PayloadBigFile . . . 56
6.112 ARex::PayloadFile Class Reference . . . 56
6.112.1 Detailed Description . . . 57
6.112.2 Constructor & Destructor Documentation . . . 57
6.112.2.1 PayloadFile . . . 57

6.112.2.2 ~PayloadFile . . . 57
6.113 Hopi::PayloadFile Class Reference . . . 57
6.113.1 Detailed Description . . . 57
6.113.2 Constructor & Destructor Documentation . . . 57
6.113.2.1 PayloadFile . . . 57
6.113.2.2 ~PayloadFile . . . 58
6.114 DREService::PerlProcessor Class Reference . . . 58
6.114.1 Constructor & Destructor Documentation . . . 58
6.114.1.1 PerlProcessor . . . 58
6.114.1.2 ~PerlProcessor . . . 58
6.115 Permission Class Reference . . . 58
6.116 PermissionGACL Class Reference . . . 59
6.117 Policy Class Reference . . . 59
6.118 ArcSec::Charon::PolicyLocation Class Reference . . . 59
6.119 GridScheduler::Resource Class Reference . . . 59
6.120 GridScheduler::ResourcesHandling Class Reference . . . 59
6.121 ContinuationPlugins::result_t Class Reference . . . 59
6.122 RTE Class Reference . . . 60
6.122.1 Detailed Description . . . 60
6.122.2 Constructor & Destructor Documentation . . . 60
6.122.2.1 RTE . . . 60
6.122.2.2 ~RTE . . . 60
6.122.3 Member Function Documentation . . . 60
6.122.3.1 Name . . . 60
6.122.3.2 operator!= . . . 60
6.122.3.3 operator< . . . 60
6.122.3.4 operator<= . . . 61
6.122.3.5 operator== . . . 61
6.122.3.6 operator> . . . 61
6.122.3.7 operator>= . . . 61
6.122.3.8 str . . . 61
6.122.3.9 Version . . . 61
6.123 RunFunction Class Reference . . . 61
6.124 RunParallel Class Reference . . . 61
6.125 gridftpd::RunPlugin Class Reference . . . 61
6.126 RunPlugin Class Reference . . . 62

6.127 RunPlugins Class Reference . . . 62
6.128 RunRedirected Class Reference . . . 62
6.129 Server Class Reference . . . 62
6.130 FileRoot::ServerParams Class Reference . . . 62
6.131 ArcSec::Service_AA Class Reference . . . 62
6.131.1 Detailed Description . . . 63
6.132 Compiler::Service_Compiler Class Reference . . . 63
6.132.1 Detailed Description . . . 63
6.132.2 Constructor & Destructor Documentation . . . 63
6.132.2.1 Service_Compiler . . . 63
6.132.3 Member Function Documentation . . . 63
6.132.3.1 process . . . 63
6.133 ArcSec::Service_Delegation Class Reference . . . 63
6.133.1 Detailed Description . . . 64
6.134 Arc::Service_JavaWrapper Class Reference . . . 64
6.134.1 Member Function Documentation . . . 64
6.134.1.1 process . . . 64
6.135 Arc::Service_PythonWrapper Class Reference . . . 64
6.135.1 Member Function Documentation . . . 64
6.135.1.1 process . . . 64
6.136 ArcSec::Service_SLCS Class Reference . . . 64
6.136.1 Detailed Description . . . 65
6.137 SPService::Service_SP Class Reference . . . 65
6.137.1 Detailed Description . . . 65
6.137.2 Constructor & Destructor Documentation . . . 65
6.137.2.1 Service_SP . . . 65
6.137.3 Member Function Documentation . . . 65
6.137.3.1 process . . . 65
6.138 SimpleMap Class Reference . . . 66
6.139 Paul::SysInfo Class Reference . . . 66
6.140 DREService::Task Class Reference . . . 66
6.140.1 Constructor & Destructor Documentation . . . 66
6.140.1.1 Task . . . 66
6.140.1.2 ∼Task . . . 66
6.141 DREService::TaskQueue Class Reference . . . 66
6.141.1 Constructor & Destructor Documentation . . . 67

6.141.1.1 TaskQueue . . . 67
6.141.1.2 ~TaskQueue . . . 67
6.141.2 Member Function Documentation . . . 67
6.141.2.1 pushTask . . . 67
6.141.2.2 shiftTask . . . 67
6.142 DREService::TaskSet Class Reference . . . 67
6.142.1 Constructor & Destructor Documentation . . . 67
6.142.1.1 TaskSet . . . 67
6.142.1.2 ~TaskSet . . . 67
6.142.2 Member Function Documentation . . . 67
6.142.2.1 removeTask . . . 67
6.143 UnixMap Class Reference . . . 68
6.144 gridftpd::UrlMapConfig Class Reference . . . 68
6.145 UrlMapConfig Class Reference . . . 68
6.146 userspec_t Class Reference . . . 68
6.147 value_for_shell Class Reference . . . 68
6.148 voms Struct Reference . . . 68
6.148.1 Detailed Description . . . 69
6.148.2 Field Documentation . . . 69
6.148.2.1 server . . . 69
6.148.2.2 std . . . 69
6.148.2.3 voname . . . 69
6.149 voms_attrs Struct Reference . . . 69
6.149.1 Detailed Description . . . 69
6.149.2 Field Documentation . . . 69
6.149.2.1 cap . . . 69
6.149.2.2 group . . . 70
6.149.2.3 role . . . 70
6.150 ARex::XMLConfig Class Reference . . . 70
6.150.1 Detailed Description . . . 70
6.150.2 Member Function Documentation . . . 70
6.150.2.1 Read . . . 70
6.150.2.2 Write . . . 70
6.151 ZeroUInt Class Reference . . . 71
6.151.1 Detailed Description . . . 71

7 File Documentation **73**

7.1 configcore.h File Reference 73
7.1.1 Detailed Description 73

Chapter 1

Namespace Index

1.1 Namespace List

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

DREService 15

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

ARex2::ARex2Service . . . 17
ARex::ARexGMConfig . . . 17
ARex::ARexJob . . . 17
ARex::ARexService . . . 20
AuthEvaluator . . . 20
AuthUser . . . 20
AuthVO . . . 21
CacheConfig . . . 21
CacheConfigException . . . 22
Cache::CacheService . . . 22
ArcSec::Charon . . . 24
CommFIFO . . . 24
Arc::Compiler_AREXClient . . . 25
Arc::Compiler_AREXClientError . . . 27
Arc::Compiler_AREXFile . . . 28
ARex::ConfGrp . . . 28
ARex::Config . . . 28
ARex::ConfigError . . . 29
ARex::ConfigIO . . . 29
    ARex::NGConfig . . . 53
    ARex::XMLConfig . . . 70
ConfigSections . . . 30
gridftpd::ConfigSections . . . 30
Paul::Configurator . . . 31
ContinuationPlugins . . . 31
ARex::CountedResource . . . 31
gridftpd::Daemon . . . 31
DirectAccess::diraccess_t . . . 31
DirectAccess . . . 31
DirEntry . . . 32
DREService::DREWebService . . . 32
Entry . . . 34
ARex::FileChunks . . . 34

ARex::FileChunksList . . . 34
FileData . . . 35
FileNode . . . 35
FilePlugin . . . 35
DirectFilePlugin . . . 32
GACLPlugin . . . 36
JobPlugin . . . 47
FileRoot . . . 36
fs_usage . . . 36
gm_dirs_ . . . 37
GMEnvironment . . . 37
gridftpd::GMEnvironment . . . 37
GridFTP_Commands . . . 38
GridFTP_Commands_timeout . . . 38
ARex::GridManager . . . 38
GridScheduler::GridSchedulerService . . . 38
Hopi::Hopi . . . 38
Paul::HTMLRequest . . . 39
Paul::HTMLResponse . . . 39
Identity . . . 39
IdentityGACL . . . 39
Index . . . 40
ARex2::JobDescription::InputFile . . . 40
Paul::InvalidMessageException . . . 41
ISIS::ISIService . . . 41
ISIS::ISISSecAttr . . . 41
Identity::Item . . . 41
IdentityItemDN . . . 40
IdentityItemVOMS . . . 40
ObjectAccess::Item . . . 41
Janitor . . . 42
Paul::Job . . . 43
Arc::Job . . . 43
Job . . . 43
job_state_rec_t . . . 45
ARex2::JobControl . . . 45
ARex2::JobController . . . 45
ARex2::JobDataCache . . . 45
JobDescription . . . 45
ARex2::JobDescription . . . 46
Paul::JobList . . . 46
JobLocalDescription . . . 46
JobLog . . . 46
ARex2::JobLRMSInfo . . . 47
Arc::JobNotFoundException . . . 47
Paul::JobQueue . . . 47
Arc::JobQueue . . . 48
Arc::JobQueueIterator . . . 48
ARex::JobRecord . . . 48
Paul::JobRequest . . . 48
Arc::JobRequest . . . 48
Arc::JobSchedMetaData . . . 48
Paul::JobSchedMetaData . . . 48

Arc::JobSelector . . . 49
JobsList . . . 49
JobsListConfig . . . 49
ARex2::JobState . . . 49
JobUser . . . 49
ARex2::JobUser . . . 50
JobUserHelper . . . 50
JobUsers . . . 50
gridftpd::LdapQuery . . . 50
gridftpd::LdapQueryError . . . 51
RunPlugin::lib_plugin_t . . . 52
gridftpd::RunPlugin::lib_plugin_t . . . 52
ARex::LoggerClient . . . 52
LRMSResult . . . 52
ISIS::Neighbor_Container . . . 53
ARex2::JobDescription::Notification . . . 53
numvalue_for_shell . . . 54
ObjectAccess . . . 54
    ObjectAccessGACL . . . 54
ARex::OptimizedInformationContainer . . . 55
ARex::Option . . . 55
ARex2::JobDescription::OutputFile . . . 55
gridftpd::ParallelLdapQueries . . . 55
Paul::PaulService . . . 55
ARex::PayloadBigFile . . . 56
Hopi::PayloadBigFile . . . 56
ARex::PayloadFile . . . 56
Hopi::PayloadFile . . . 57
DREService::PerlProcessor . . . 58
Permission . . . 58
    PermissionGACL . . . 59
Policy . . . 59
ArcSec::Charon::PolicyLocation . . . 59
GridScheduler::Resource . . . 59
GridScheduler::ResourcesHandling . . . 59
ContinuationPlugins::result_t . . . 59
RTE . . . 60
RunFunction . . . 61
RunParallel . . . 61
gridftpd::RunPlugin . . . 61
RunPlugin . . . 62
RunPlugins . . . 62
RunRedirected . . . 62
Server . . . 62
FileRoot::ServerParams . . . 62
ArcSec::Service_AA . . . 62
Compiler::Service_Compiler . . . 63
ArcSec::Service_Delegation . . . 63
Arc::Service_JavaWrapper . . . 64
Arc::Service_PythonWrapper . . . 64
ArcSec::Service_SLCS . . . 64
SPService::Service_SP . . . 65
SimpleMap . . . 66

Paul::SysInfo 66
DREService::Task 66
DREService::TaskQueue 66
DREService::TaskSet 67
UnixMap 68
gridftpd::UrlMapConfig 68
UrlMapConfig 68
userspec_t 68
value_for_shell 68
voms 68
voms_attrs 69
ZeroUInt 71

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

ARex2::ARex2Service . . . 17
ARex::ARexGMConfig . . . 17
ARex::ARexJob . . . 17
ARex::ARexService . . . 20
AuthEvaluator . . . 20
AuthUser . . . 20
AuthVO . . . 21
CacheConfig . . . 21
CacheConfigException . . . 22
Cache::CacheService . . . 22
ArcSec::Charon . . . 24
CommFIFO . . . 24
Arc::Compiler_AREXClient (A client class for the A-REX service) . . . 25
Arc::Compiler_AREXClientError (An exception class for the **Compiler_AREXClient** (p. 25) class) . . . 27
Arc::Compiler_AREXFile . . . 28
ARex::ConfGrp . . . 28
ARex::Config . . . 28
ARex::ConfigError . . . 29
ARex::ConfigIO . . . 29
ConfigSections . . . 30
gridftpd::ConfigSections . . . 30
Paul::Configurator . . . 31
ContinuationPlugins . . . 31
ARex::CountedResource . . . 31
gridftpd::Daemon . . . 31
DirectAccess::diraccess_t . . . 31
DirectAccess . . . 31
DirectFilePlugin . . . 32
DirEntry . . . 32
DREService::DREWebService . . . 32
Entry . . . 34
ARex::FileChunks (Representation of delivered file chunks) . . . 34

ARex::FileChunksList (Container for **FileChunks** (p. 34) instances) . . . 34
FileData . . . 35
FileNode . . . 35
FilePlugin . . . 35
FileRoot . . . 36
fs_usage . . . 36
GACLPlugin . . . 36
gm_dirs_ . . . 37
GMEnvironment . . . 37
gridftpd::GMEnvironment . . . 37
GridFTP_Commands . . . 38
GridFTP_Commands_timeout . . . 38
ARex::GridManager . . . 38
GridScheduler::GridSchedulerService . . . 38
Hopi::Hopi . . . 38
Paul::HTMLRequest . . . 39
Paul::HTMLResponse . . . 39
Identity . . . 39
IdentityGACL . . . 39
IdentityItemDN . . . 40
IdentityItemVOMS . . . 40
Index . . . 40
ARex2::JobDescription::InputFile . . . 40
Paul::InvalidMessageException . . . 41
ISIS::ISIService . . . 41
ISIS::ISISSecAttr . . . 41
Identity::Item . . . 41
ObjectAccess::Item . . . 41
Janitor (Class to communicate with **Janitor** (p. 42) - Dynmaic Runtime Environment handler) . 42
Paul::Job . . . 43
Arc::Job . . . 43
Job . . . 43
job_state_rec_t . . . 45
ARex2::JobControl . . . 45
ARex2::JobController . . . 45
ARex2::JobDataCache . . . 45
JobDescription . . . 45
ARex2::JobDescription . . . 46
Paul::JobList . . . 46
JobLocalDescription . . . 46
JobLog . . . 46
ARex2::JobLRMSInfo . . . 47
Arc::JobNotFoundException . . . 47
JobPlugin . . . 47
Paul::JobQueue . . . 47
Arc::JobQueue . . . 48
Arc::JobQueueIterator . . . 48
ARex::JobRecord . . . 48
Paul::JobRequest . . . 48
Arc::JobRequest . . . 48
Arc::JobSchedMetaData . . . 48
Paul::JobSchedMetaData . . . 48
Arc::JobSelector . . . 49
JobsList . . . 49

JobsListConfig . . . 49
ARex2::JobState . . . 49
JobUser . . . 49
ARex2::JobUser . . . 50
JobUserHelper . . . 50
JobUsers . . . 50
gridftpd::LdapQuery . . . 50
gridftpd::LdapQueryError . . . 51
RunPlugin::lib_plugin_t . . . 52
gridftpd::RunPlugin::lib_plugin_t . . . 52
ARex::LoggerClient . . . 52
LRMSResult . . . 52
ISIS::Neighbor_Container . . . 53
ARex::NGConfig . . . 53
ARex2::JobDescription::Notification . . . 53
numvalue_for_shell . . . 54
ObjectAccess . . . 54
ObjectAccessGACL . . . 54
ARex::OptimizedInformationContainer . . . 55
ARex::Option . . . 55
ARex2::JobDescription::OutputFile . . . 55
gridftpd::ParallelLdapQueries . . . 55
Paul::PaulService . . . 55
ARex::PayloadBigFile . . . 56
Hopi::PayloadBigFile . . . 56
ARex::PayloadFile . . . 56
Hopi::PayloadFile . . . 57
DREService::PerlProcessor . . . 58
Permission . . . 58
PermissionGACL . . . 59
Policy . . . 59
ArcSec::Charon::PolicyLocation . . . 59
GridScheduler::Resource . . . 59
GridScheduler::ResourcesHandling . . . 59
ContinuationPlugins::result_t . . . 59
RTE . . . 60
RunFunction . . . 61
RunParallel . . . 61
gridftpd::RunPlugin . . . 61
RunPlugin . . . 62
RunPlugins . . . 62
RunRedirected . . . 62
Server . . . 62
FileRoot::ServerParams . . . 62
ArcSec::Service_AA . . . 62
Compiler::Service_Compiler . . . 63
ArcSec::Service_Delegation (A Service which launches the proxy certificate request; it accepts the request from) . . . 63
Arc::Service_JavaWrapper . . . 64
Arc::Service_PythonWrapper . . . 64
ArcSec::Service_SLCS . . . 64
SPService::Service_SP . . . 65
SimpleMap . . . 66
Paul::SysInfo . . . 66

DREService::Task 66
DREService::TaskQueue 66
DREService::TaskSet 67
UnixMap 68
gridftpd::UrlMapConfig 68
UrlMapConfig 68
userspec_t 68
value_for_shell 68
voms 68
voms_attrs 69
ARex::XMLConfig 70
ZeroUInt 71

Chapter 4

File Index

4.1 File List

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

aaservice.h . . . ??
arex.h . . . ??
arex2.h . . . ??
arex_client.h . . . ??
auth.h . . . ??
CacheService.h . . . ??
a-rex/grid-manager/misc/canonical_dir.h . . . ??
gridftpd/misc/canonical_dir.h . . . ??
charon.h . . . ??
client.h . . . ??
commands.h . . . ??
commfifo.h . . . ??
compiler.h . . . ??
a-rex/grid-manager/conf/conf.h . . . ??
gridftpd/conf/conf.h . . . ??
gridftpd/conf.h . . . ??
conf_cache.h . . . ??
conf_file.h . . . ??
a-rex/grid-manager/conf/conf_map.h . . . ??
gridftpd/conf/conf_map.h . . . ??
conf_pre.h . . . ??
a-rex/grid-manager/conf/conf_sections.h . . . ??
gridftpd/conf/conf_sections.h . . . ??
conf_vo.h . . . ??
configcore.h . . . 73
configio.h . . . ??
configurator.h . . . ??
daemon.h . . . ??
delegation.h . . . ??
delete.h . . . ??
dREWebService.h . . . ??
Entry.h . . . ??
a-rex/grid-manager/conf/environment.h . . . ??

gridftpd/conf/environment.h . . . ??
a-rex/grid-manager/misc/escaped.h . . . ??
gridftpd/misc/escaped.h . . . ??
FileChunks.h . . . ??
fileplugin.h . . . ??
fileroot.h . . . ??
fsusage.h . . . ??
a-rex/gacl.h . . . ??
gridftpd/external/gacl/gacl.h . . . ??
gacl_auth.h . . . ??
gaclplugin.h . . . ??
grid_manager.h . . . ??
grid_sched.h . . . ??
a-rex/grid-manager/conf/gridmap.h . . . ??
gridftpd/conf/gridmap.h . . . ??
hopi.h . . . ??
identity.h . . . ??
identity_dn.h . . . ??
identity_gacl.h . . . ??
identity_voms.h . . . ??
Index.h . . . ??
info_files.h . . . ??
info_log.h . . . ??
info_types.h . . . ??
isis.h . . . ??
janitor.h . . . ??
javawrapper.h . . . ??
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_desc.h . . . ??
job_descr.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 . . . ??
jobplugin.h . . . ??
JobRecord.h . . . ??
ldapquery.h . . . ??
LDIFtoXML.h . . . ??
lrms.h . . . ??

misc.h . . . ??
names.h . . . ??
ngconfig.h . . . ??
object_access.h . . . ??
object_access_gacl.h . . . ??
paul.h . . . ??
a-rex/PayloadFile.h . . . ??
hopi/PayloadFile.h . . . ??
PerlProcessor.h . . . ??
permission.h . . . ??
permission_gacl.h . . . ??
plugins.h . . . ??
Policy.h . . . ??
a-rex/grid-manager/misc/proxy.h . . . ??
gridftpd/misc/proxy.h . . . ??
pythonwrapper.h . . . ??
resource.h . . . ??
resources_handling.h . . . ??
rte.h . . . ??
run_function.h . . . ??
run_parallel.h . . . ??
a-rex/grid-manager/run/run_plugin.h . . . ??
gridftpd/run/run_plugin.h . . . ??
run_redirected.h . . . ??
security.h . . . ??
send_mail.h . . . ??
Server.h . . . ??
simplemap.h . . . ??
slcs.h . . . ??
SPService.h . . . ??
states.h . . . ??
sysinfo.h . . . ??
Task.h . . . ??
TaskQueue.h . . . ??
TaskSet.h . . . ??
tools.h . . . ??
unixmap.h . . . ??
users.h . . . ??
userspec.h . . . ??
util++.h . . . ??
util.h . . . ??
xmlconfig.h . . . ??

Chapter 5

Namespace Documentation

5.1 DREService Namespace Reference

Data Structures

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

5.1.1 Detailed Description

Implementation of a simple echo service

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

Chapter 6

Data Structure Documentation

6.1 ARex2::ARex2Service Class Reference

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

- arex2.h

6.2 ARex::ARexGMConfig Class Reference

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

- a-rex/job.h

6.3 ARex::ARexJob Class Reference

`#include <job.h>`

Public Member Functions

- **ARexJob** (const std::string &id, **ARexGMConfig** &config, Arc::Logger &logger, bool fast_auth_-check=false)
- **ARexJob** (Arc::XMLNode jsdl, **ARexGMConfig** &config, const std::string &credentials, const std::string &clientid, Arc::Logger &logger, Arc::XMLNode migration=Arc::XMLNode())
- std::string **Failure** (void)
- std::string **ID** (void)
- bool **GetDescription** (Arc::XMLNode &jsdl)
- bool **Cancel** (void)
- bool **Clean** (void)
- bool **Resume** (void)
- std::string **State** (void)
- std::string **State** (bool &job_pending)
- bool **Failed** (void)
- std::string **SessionDir** (void)

- std::string **LogDir** (void)
- int **CreateFile** (const std::string &filename)
- int **OpenFile** (const std::string &filename, bool for_read, bool for_write)
- int **OpenLogFile** (const std::string &name)
- Glib::Dir * **OpenDir** (const std::string &dirname)
- std::list< std::string > **LogFiles** (void)
- bool **UpdateCredentials** (const std::string &credentials)
- bool **ChooseSessionDir** (const std::string &jobid, std::string &sessiondir)

Static Public Member Functions

- static int **TotalJobs** (**ARexGMConfig** &config, Arc::Logger &logger)
- static std::list< std::string > **Jobs** (**ARexGMConfig** &config, Arc::Logger &logger)

6.3.1 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.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 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

6.3.2.2 ARex::ARexJob::ARexJob (Arc::XMLNode *jsdl,* ARexGMConfig & *config,* const std::string & *credentials,* const std::string & *clientid,* Arc::Logger & *logger,* Arc::XMLNode *migration* = `Arc::XMLNode()`)

Create new job with provided JSDL description

6.3.3 Member Function Documentation

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

Cancel processing/execution of job

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

Select a session dir to use for this job

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

Remove job from local pool

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

Creates file in job's session directory and returns handler

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

Returns true if job has failed

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

Returns textual description of failure of last operation

6.3.3.7 bool ARex::ARexJob::GetDescription (Arc::XMLNode & *jsdl*)

Fills provided jsdl with job description

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

Return ID assigned to job

6.3.3.9 static std::list<std::string> ARex::ARexJob::Jobs (ARexGMConfig & *config*, Arc::Logger & *logger*) [static]

Returns list of user's jobs. Fine-grained ACL is ignored.

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

Returns name of virtual log directory

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

Returns list of existing log files

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

Opens directory inside session directory

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

Opens file in job's session directory and returns handler

6.3.3.14 int ARex::ARexJob::OpenLogFile (const std::string & *name*)

Opens log file in control directory

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

Resume execution of job after error

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

Returns path to session directory

6.3.3.17 std::string ARex::ARexJob::State (bool & *job_pending*)

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

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

Returns current state of job

6.3.3.19 static int ARex::ARexJob::TotalJobs (ARexGMConfig & *config*, Arc::Logger & *logger*) [static]

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

6.3.3.20 bool ARex::ARexJob::UpdateCredentials (const std::string & *credentials*)

Updates job credentials

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

- a-rex/job.h

6.4 ARex::ARexService Class Reference

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

- arex.h

6.5 AuthEvaluator Class Reference

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

- auth.h

6.6 AuthUser Class Reference

Data Structures

- class **group_t**

- struct **source_t**

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

- auth.h

6.7 AuthVO Class Reference

Friends

- class **AuthUser**

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

- auth.h

6.8 CacheConfig Class Reference

`#include <conf_cache.h>`

Public Member Functions

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

6.8.1 Detailed Description

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

6.8.2 Constructor & Destructor Documentation

6.8.2.1 CacheConfig::CacheConfig (const GMEnvironment & *env,* std::string *username* = "")

Create a new **CacheConfig** (p. 21) instance. Read the config file and fill in private member variables with cache parameters. If different users are defined in the conf file, use the cache parameters for the given username.

6.8.2.2 CacheConfig::CacheConfig ( ) [inline]

Empty **CacheConfig** (p. 21)

6.8.3 Member Function Documentation

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

Parsers for the two different conf styles

6.8.3.2 void CacheConfig::setCacheDirs (std::vector< std::string > *cache_dirs*) [inline]

To allow for substitutions done during configuration

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

- conf_cache.h

6.9 CacheConfigException Class Reference

`#include <conf_cache.h>`

6.9.1 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

6.10 Cache::CacheService Class Reference

`#include <CacheService.h>`

Public Member Functions

- **CacheService** (Arc::Config *cfg)
- virtual ~**CacheService** (void)
- virtual Arc::MCC_Status **process** (Arc::Message &inmsg, Arc::Message &outmsg)
- bool **RegistrationCollector** (Arc::XMLNode &doc)
- **operator bool** ()
- bool **operator!** ()

Protected Member Functions

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

6.10.1 Detailed Description

CacheService (p. 22) provides functionality for A-REX cache operations that can be performed by remote clients. It currently consists of two operations: CacheCheck - allows querying of the cache for the presence of files. CacheLink - enables a running job to dynamically request cache files to be linked to its working (session) directory. This is especially useful in the case of pilot job workflows where job submission does not follow the usual ARC workflow. In order for input files to be available to jobs, the pilot job can call the cache service to prepare them. If requested files are not present in the cache, they can be downloaded by the cache service if requested, using the A-REX downloader utility.

6.10.2 Constructor & Destructor Documentation

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

Make a new **CacheService** (p. 22). Reads the configuration and determines the validity of the service.

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

Destroy the **CacheService** (p. 22)

6.10.3 Member Function Documentation

6.10.3.1 Arc::MCC_Status Cache::CacheService::CacheCheck (Arc::XMLNode *in*, Arc::XMLNode *out*, const JobUser & *user*) [protected]

Check whether the URLs supplied in the input are present in any cache. Returns in the out message for each file true or false, and if true, the size of the file on cache disk.

Parameters

user A-REX user configuration for the mapped user

6.10.3.2 Arc::MCC_Status Cache::CacheService::CacheLink (Arc::XMLNode *in*, Arc::XMLNode *out*, const JobUser & *user*, const Arc::User & *mapped_user*) [protected]

This method is used to link cache files to the session dir. A list of URLs is supplied and if they are present in the cache and the user calling the service has permission to access them, then they are linked to the given session directory. If the user requests that missing files be staged, then a downloader process is launched to obtain them.

Parameters

user A-REX user configuration for the mapped user

mapped_user The local user to which the client DN was mapped

6.10.3.3 Cache::CacheService::operator bool (void) [inline]

Returns true if the **CacheService** (p. 22) is valid.

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

Returns true if the **CacheService** (p. 22) is not valid.

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

Main method called by HED when **CacheService** (p. 22) is invoked. Directs call to appropriate **CacheService** (p. 22) method.

6.10.3.6 bool Cache::CacheService::RegistrationCollector (Arc::XMLNode & *doc*)

Supplies information on the service for use in the information system.

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

- CacheService.h

6.11 ArcSec::Charon Class Reference

`#include <charon.h>`

Data Structures

- class **PolicyLocation**

6.11.1 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

6.12 CommFIFO Class Reference

Data Structures

- class **elem_t**

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

- commfifo.h

6.13 Arc::Compiler_AREXClient Class Reference

A client class for the A-REX service.

`#include <arex_client.h>`

Public Member Functions

- **Compiler_AREXClient** (std::string configFile="") throw (Compiler_AREXClientError)
- ~**Compiler_AREXClient** ()
- std::string **submit** (std::istream &jsdl_file, Compiler_AREXFileList &file_list, bool delegate=false) throw (Compiler_AREXClientError)
- std::string **stat** (const std::string &jobid) throw (Compiler_AREXClientError)
- void **kill** (const std::string &jobid) throw (Compiler_AREXClientError)
- void **clean** (const std::string &jobid) throw (Compiler_AREXClientError)
- std::string **sstat** (void) throw (Compiler_AREXClientError)

6.13.1 Detailed Description

A client class for the A-REX service. This class is a client for the A-REX service (Arc Resource-coupled EXecution service). It provides methods for three operations on an A-REX service:

- **Job** (p. 43) submission
- **Job** (p. 43) status queries
- **Job** (p. 43) termination

6.13.2 Constructor & Destructor Documentation

6.13.2.1 Arc::Compiler_AREXClient::Compiler_AREXClient (std::string *configFile* = "") throw (Compiler_AREXClientError)

The constructor for the **Compiler_AREXClient** (p. 25) class.

This is the constructor for the **Compiler_AREXClient** (p. 25) class. It creates an A-REX client that corresponds to a specific A-REX service, which is specified in a configuration file. The configuration file also specifies how to set up the communication chain for the client. The location of the configuration file can be provided as a parameter to the method. If no such parameter is given, the environment variable ARC_-Compiler_AREX_CONFIG is assumed to contain the location. If there is no such environment variable, the configuration file is assumed to be "arex_client.xml" in the current working directory.

Parameters

configFile The location of the configuration file.

Exceptions

An **Compiler_AREXClientError** (p. 27) object if an error occurs.

6.13.2.2 Arc::Compiler_AREXClient::~Compiler_AREXClient ( )

The destructor.

This is the destructor. It does what destructors usually do, cleans up...

6.13.3 Member Function Documentation

6.13.3.1 void Arc::Compiler_AREXClient::clean (const std::string & *jobid*) throw (Compiler_AREXClientError)

Removes a job.

This method sends a request to the A-REX service to remove a job from it's pool. If job is running it will be killed by service as well.

Parameters

jobid The **Job** (p. 43) ID of the job to remove.

Exceptions

An **Compiler_AREXClientError** (p. 27) object if an error occurs.

6.13.3.2 void Arc::Compiler_AREXClient::kill (const std::string & *jobid*) throw (Compiler_AREXClientError)

Terminates a job.

This method sends a request to the A-REX service to terminate a job.

Parameters

jobid The **Job** (p. 43) ID of the job to terminate.

Exceptions

An **Compiler_AREXClientError** (p. 27) object if an error occurs.

6.13.3.3 std::string Arc::Compiler_AREXClient::sstat (void) throw (Compiler_AREXClientError)

Query the status of a service.

This method queries the A-REX service about it's status.

Returns

The XML document representing status of the service.

Exceptions

An **Compiler_AREXClientError** (p. 27) object if an error occurs.

6.13.3.4 std::string Arc::Compiler_AREXClient::stat (const std::string & *jobid*) throw (Compiler_AREXClientError)

Query the status of a job.

This method queries the A-REX service about the status of a job.

Parameters

jobid The **Job** (p. 43) ID of the job.

Returns

The status of the job.

Exceptions

An **Compiler_AREXClientError** (p. 27) object if an error occurs.

6.13.3.5 std::string Arc::Compiler_AREXClient::submit (std::istream & *jsdl_file*, Compiler_AREXFileList & *file_list*, bool *delegate* = `false`) throw (Compiler_AREXClientError)

Submit a job.

This method submits a job to the A-REX service corresponding to this client instance.

Parameters

jsdl_file An input stream from which the JSDL file for the job can be read.

Returns

The **Job** (p. 43) ID of the the submitted job.

Exceptions

An **Compiler_AREXClientError** (p. 27) object if an error occurs.

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

- arex_client.h

6.14 Arc::Compiler_AREXClientError Class Reference

An exception class for the **Compiler_AREXClient** (p. 25) class.

`#include <arex_client.h>`

Public Member Functions

- **Compiler_AREXClientError** (const std::string &what="")

6.14.1 Detailed Description

An exception class for the **Compiler_AREXClient** (p. 25) class. This is an exception class that is used to handle runtime errors discovered in the **Compiler_AREXClient** (p. 25) class.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 Arc::Compiler_AREXClientError::Compiler_AREXClientError (const std::string & *what* = "")

Constructor.

This is the constructor of the **Compiler_AREXClientError** (p. 27) class.

Parameters

what An explanation of the error.

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

- arex_client.h

6.15 Arc::Compiler_AREXFile Class Reference

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

- arex_client.h

6.16 ARex::ConfGrp Class Reference

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

- **configcore.h**

6.17 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

6.17.1 Detailed Description

Core configuration class.

6.17.2 Member Function Documentation

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

Get the configuration values from key.

6.17.2.2 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.

6.17.2.3 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**

6.18 ARex::ConfigError Class Reference

`#include <configcore.h>`

Public Member Functions

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

6.18.1 Detailed Description

Error configuration class.

6.18.2 Constructor & Destructor Documentation

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

Constructor for the **ConfigError** (p. 29) exception. Calls the corresponding constructor in ARCLibError.

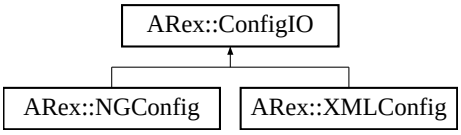
The documentation for this class was generated from the following file:

- **configcore.h**

6.19 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

6.19.1 Detailed Description

Virtual base-class for reading and writing configuration files. Concrete instances include **NGConfig** (p. 53) and **XMLConfig** (p. 70).

6.19.2 Member Function Documentation

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

Read the named configuration source.

Implemented in **ARex::NGConfig** (p. 53), and **ARex::XMLConfig** (p. 70).

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

Write configuration to named configuration destination.

Implemented in **ARex::NGConfig** (p. 53), and **ARex::XMLConfig** (p. 70).

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

- configio.h

6.20 ConfigSections Class Reference

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

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

6.21 gridftpd::ConfigSections Class Reference

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

- gridftpd/conf/conf_sections.h

6.22 Paul::Configurator Class Reference

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

- configurator.h

6.23 ContinuationPlugins Class Reference

Data Structures

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

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

- plugins.h

6.24 ARex::CountedResource Class Reference

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

- arex.h

6.25 gridftpd::Daemon Class Reference

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

- daemon.h

6.26 DirectAccess::diraccess_t Struct Reference

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

- fileplugin.h

6.27 DirectAccess Class Reference

Data Structures

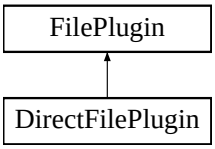
- struct **diraccess_t**

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

- fileplugin.h

6.28 DirectFilePlugin Class Reference

Inheritance diagram for DirectFilePlugin:



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

- fileplugin.h

6.29 DirEntry Class Reference

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

- fileroot.h

6.30 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**

6.30.1 Constructor & Destructor Documentation

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

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

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

Destructor.

6.30.2 Member Function Documentation

6.30.2.1 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

6.30.2.2 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

6.30.3 Field Documentation

6.30.3.1 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.

6.30.3.2 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

6.31 Entry Class Reference

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

- Entry.h

6.32 ARex::FileChunks Class Reference

Representation of delivered file chunks.

`#include <FileChunks.h>`

Public Member Functions

- std::string **Path** (void)
- void **Size** (off_t size)
- off_t **Size** (void)
- void **Add** (off_t start, size_t csize)
- bool **Complete** (void)
- void **Print** (void)
- void **Release** (void)
- void **Remove** (void)

6.32.1 Detailed Description

Representation of delivered file chunks.

6.32.2 Member Function Documentation

6.32.2.1 void ARex::FileChunks::Release (void)

Release reference obtained through **FileChunksList::Get()** (p. 35) method. This operation may lead to destruction of FileChunk instance hence previously obtained refrence mus tnot be used.

6.32.2.2 void ARex::FileChunks::Remove (void)

Relases reference obtained through Get() method and destroys its instance. Normally this method to be called instead of **Release()** (p. 34) after whole file is delivered in order to free resources associated with **FileChunks** (p. 34) instance.

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

- FileChunks.h

6.33 ARex::FileChunksList Class Reference

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

`#include <FileChunks.h>`

Public Member Functions

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

6.33.1 Detailed Description

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

6.33.2 Member Function Documentation

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

Returns previously created **FileChunks** (p. 34) object with associated path. If such instance does not exist new one is created. Obtained reference may be used for other operations. Obtained reference must be Release()ed after it is not longer needed.

6.33.2.2 FileChunks∗ ARex::FileChunksList::GetStuck (void)

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

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

- FileChunks.h

6.34 FileData Class Reference

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

- info_types.h

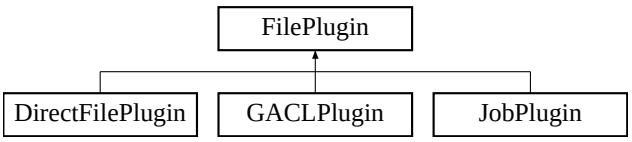
6.35 FileNode Class Reference

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

- fileroot.h

6.36 FilePlugin Class Reference

Inheritance diagram for FilePlugin:



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

- fileroot.h

6.37 FileRoot Class Reference

Data Structures

- class **ServerParams**

Friends

- class **GridFTP_Commands**

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

- fileroot.h

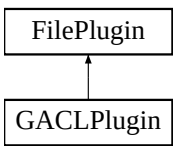
6.38 fs_usage Struct Reference

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

- fsusage.h

6.39 GACLPlugin Class Reference

Inheritance diagram for GACLPlugin:



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

- gaclplugin.h

6.40 gm_dirs_ Struct Reference

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

- jobplugin.h

6.41 GMEnvironment Class Reference

Public Member Functions

- std::string **nordugrid_loc** (void) const
- std::string **nordugrid_libexec_loc** (void) const
- std::string **nordugrid_config_loc** (void) const
- std::string **support_mail_address** (void) const

6.41.1 Member Function Documentation

6.41.1.1 std::string GMEnvironment::nordugrid_config_loc (void) const

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

6.41.1.2 std::string GMEnvironment::support_mail_address (void) const

Email address of person responsible for this ARC installation grid.manager, it can also be set from configuration file

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

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

6.42 gridftpd::GMEnvironment Class Reference

Public Member Functions

- std::string **nordugrid_loc** (void) const
- std::string **nordugrid_libexec_loc** (void) const
- std::string **nordugrid_config_loc** (void) const
- std::string **support_mail_address** (void) const

6.42.1 Member Function Documentation

6.42.1.1 std::string gridftpd::GMEnvironment::nordugrid_config_loc (void) const

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

6.42.1.2 std::string gridftpd::GMEnvironment::support_mail_address (void) const

Email address of person responsible for this ARC installation grid.manager, it can also be set from configuration file

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

- gridftpd/conf/environment.h

6.43 GridFTP_Commands Class Reference

Data Structures

- class **close_semaphor_t**
- struct **data_buffer_t**

Friends

- class **GridFTP_Commands_timeout**

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

- commands.h

6.44 GridFTP_Commands_timeout Class Reference

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

- commands.h

6.45 ARex::GridManager Class Reference

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

- grid_manager.h

6.46 GridScheduler::GridSchedulerService Class Reference

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

- grid_sched.h

6.47 Hopi::Hopi Class Reference

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

- hopi.h

6.48 Paul::HTMLRequest Class Reference

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

- configurator.h

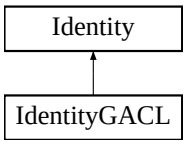
6.49 Paul::HTMLResponse Class Reference

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

- configurator.h

6.50 Identity Class Reference

Inheritance diagram for Identity:



Data Structures

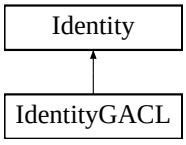
- class **Item**

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

- identity.h

6.51 IdentityGACL Class Reference

Inheritance diagram for IdentityGACL:

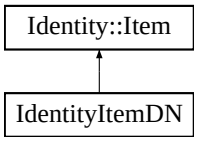


The documentation for this class was generated from the following file:

- identity_gacl.h

6.52 IdentityItemDN Class Reference

Inheritance diagram for IdentityItemDN:

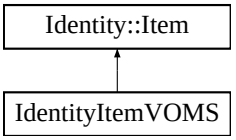


The documentation for this class was generated from the following file:

- identity_dn.h

6.53 IdentityItemVOMS Class Reference

Inheritance diagram for IdentityItemVOMS:



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

- identity_voms.h

6.54 Index Class Reference

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

- Index.h

6.55 ARex2::JobDescription::InputFile Class Reference

`#include <job_descr.h>`

6.55.1 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

6.56 Paul::InvalidMessageException Class Reference

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

- configurator.h

6.57 ISIS::ISIService Class Reference

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

- isis.h

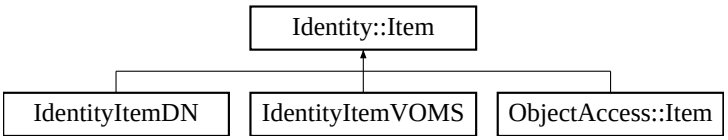
6.58 ISIS::ISISSecAttr Class Reference

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

- security.h

6.59 Identity::Item Class Reference

Inheritance diagram for Identity::Item:

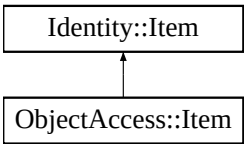


The documentation for this class was generated from the following file:

- identity.h

6.60 ObjectAccess::Item Class Reference

Inheritance diagram for ObjectAccess::Item:



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

- object_access.h

6.61 Janitor Class Reference

Class to communicate with **Janitor** (p. 42) - Dynmaic Runtime Environment handler.

`#include <janitor.h>`

Public Member Functions

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

6.61.1 Detailed Description

Class to communicate with **Janitor** (p. 42) - Dynmaic Runtime Environment handler.

6.61.2 Constructor & Destructor Documentation

6.61.2.1 Janitor::Janitor (const std::string & *id,* const std::string & *cdir,* const GMEnvironment & *env*)

Creates instance representing job entry in **Janitor** (p. 42) database.

Takes id for job identifier and cdir for the control directory of A-Rex. constructor does not register job in the **Janitor** (p. 42). It only associates job with this instance.

6.61.3 Member Function Documentation

6.61.3.1 bool Janitor::deploy (void)

Registers associated job with **Janitor** (p. 42) and deploys dynamic RTEs.

This operation is asynchronous. Returned true means **Janitor** (p. 42) will be contacted and deployemnt will start soon. For obtaining result of operation see methods **wait()** (p. 43) and **result()** (p. 43). During this operation janitor utility is called with command register and optionally deploy.

6.61.3.2 bool Janitor::remove (void)

Removes job from those handled by **Janitor** (p. 42) and releases associated RTEs.

This operation is asynchronous. Returned true means **Janitor** (p. 42) will be contacted and removal will start soon. For obtaining result of operation see methods **wait()** (p. 43) and **result()** (p. 43). During this operation janitor utility is called with command remove.

6.61.3.3 Result Janitor::result (void)

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

It should be called after **wait()** (p. 43) returned true.

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

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

This operation returns true if operation finished or false if timeout seconds passed. It may be called repeatedly and even after it previously returned true. If no operation is running it returns true immeaditely.

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

- janitor.h

6.62 Paul::Job Class Reference

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

- paul/job.h

6.63 Arc::Job Class Reference

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

- sched/job.h

6.64 Job Class Reference

`#include <job.h>`

Public Member Functions

- **Job** (void)
- **Job** (std::string path)
- ~**Job** (void)
- **operator bool** (void)
- bool **Start** (void)
- bool **Cancel** (void)
- bool **Resume** (void)
- std::string **GetState** (void)
- std::string **GetSessionDir** (void)

6.64.1 Detailed Description

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

6.64.2 Constructor & Destructor Documentation

6.64.2.1 Job::Job (void)

Constructor: Creates empty job

6.64.2.2 Job::Job (std::string *path*)

Constructor: load job information form files

6.64.2.3 Job::~Job (void)

Destuctior

6.64.3 Member Function Documentation

6.64.3.1 bool Job::Cancel (void)

Cancel processing/execution of job

6.64.3.2 std::string Job::GetSessionDir (void)

Returns the session directory of the job

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

Returns the string represnetation of job state

6.64.3.4 Job::operator bool (void) `[inline]`

Helper logical operators

6.64.3.5 bool Job::Resume (void)

Resume execution of job after error

6.64.3.6 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

6.65 job_state_rec_t Struct Reference

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

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

6.66 ARex2::JobControl Class Reference

`#include <job_control.h>`

6.66.1 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

6.67 ARex2::JobController Class Reference

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

- job_control.h

6.68 ARex2::JobDataCache Class Reference

`#include <job_data_cache.h>`

6.68.1 Detailed Description

Data cache

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

- job_data_cache.h

6.69 JobDescription Class Reference

Friends

- class **JobsList**

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

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

6.70 ARex2::JobDescription Class Reference

`#include <job_descr.h>`

Data Structures

- class **InputFile**
- class **Notification**
- class **OutputFile**

Public Member Functions

- std::string & **JobName** (void)

6.70.1 Detailed Description

Internal representation of **Job** (p. 43) described by JSDL

6.70.2 Member Function Documentation

6.70.2.1 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

6.71 Paul::JobList Class Reference

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

- job_list.h

6.72 JobLocalDescription Class Reference

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

- info_types.h

6.73 JobLog Class Reference

`#include <job_log.h>`

6.73.1 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

6.74 ARex2::JobLRMSInfo Class Reference

`#include <lrms.h>`

6.74.1 Detailed Description

Class represents the information about job in LRMS

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

- lrms.h

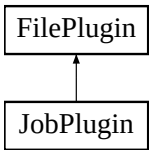
6.75 Arc::JobNotFoundException Class Reference

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

- sched/job_queue.h

6.76 JobPlugin Class Reference

Inheritance diagram for JobPlugin:



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

- jobplugin.h

6.77 Paul::JobQueue Class Reference

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

- paul/job_queue.h

6.78 Arc::JobQueue Class Reference

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

- sched/job_queue.h

6.79 Arc::JobQueueIterator Class Reference

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

- sched/job_queue.h

6.80 ARex::JobRecord Class Reference

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

- JobRecord.h

6.81 Paul::JobRequest Class Reference

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

- paul/job_request.h

6.82 Arc::JobRequest Class Reference

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

- sched/job_request.h

6.83 Arc::JobSchedMetaData Class Reference

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

- sched/job_sched_meta.h

6.84 Paul::JobSchedMetaData Class Reference

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

- paul/job_sched_meta.h

6.85 Arc::JobSelector Class Reference

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

- sched/job_queue.h

6.86 JobsList Class Reference

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

- states.h

6.87 JobsListConfig Class Reference

`#include <states.h>`

Friends

- class **JobsList**

6.87.1 Detailed Description

Class to represent information read from configuration.

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

- states.h

6.88 ARex2::JobState Class Reference

`#include <job_state.h>`

6.88.1 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

6.89 JobUser Class Reference

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

- users.h

6.90 ARex2::JobUser Class Reference

`#include <job_user.h>`

6.90.1 Detailed Description

Job (p. 43) run under the privilages 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

6.91 JobUserHelper Class Reference

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

- users.h

6.92 JobUsers Class Reference

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

- users.h

6.93 gridftpd::LdapQuery Class Reference

`#include <ldapquery.h>`

Public Types

- enum **Scope**

Public Member Functions

- **LdapQuery** (const std::string &ldaphost, int ldapport, bool anonymous=true, const std::string &usersn="", int timeout=20)
- ~**LdapQuery** ()
- void **Query** (const std::string &base, const std::string &filter="(objectclass=*)", const std::vector< std::string > &attributes=std::vector< std::string >(), **Scope** scope=subtree) throw (LdapQueryError)
- void **Result** (ldap_callback callback, void *ref) throw (LdapQueryError)
- std::string **Host** ()

6.93.1 Detailed Description

LdapQuery (p. 50) class; querying of LDAP servers.

6.93.2 Member Enumeration Documentation

6.93.2.1 enum gridftpd::LdapQuery::Scope

Scope for a LDAP queries. Use when querying.

6.93.3 Constructor & Destructor Documentation

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

Constructs a new **LdapQuery** (p. 50) object and sets connection options. The connection is first established when calling Query.

6.93.3.2 gridftpd::LdapQuery::~LdapQuery ( )

Destructor. Will disconnect from the ldapserver if stll connected.

6.93.4 Member Function Documentation

6.93.4.1 std::string gridftpd::LdapQuery::Host ( )

Returns the hostname of the ldap-server.

6.93.4.2 void gridftpd::LdapQuery::Query (const std::string & *base,* const std::string & *filter* = `"(objectclass=*)"`, const std::vector< std::string > & *attributes* = `std::vector< std::string >()`, Scope *scope* = `subtree`) throw (LdapQueryError)

Queries the ldap server.

6.93.4.3 void gridftpd::LdapQuery::Result (ldap_callback *callback,* void * *ref*) throw (LdapQueryError)

Retrieves the result of the query from the ldap-server.

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

- ldapquery.h

6.94 gridftpd::LdapQueryError Class Reference

`#include <ldapquery.h>`

Public Member Functions

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

6.94.1 Detailed Description

LdapQuery (p. 50) exception. Gets thrown whan an error occurs in a query.

6.94.2 Constructor & Destructor Documentation

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

Standard exception class constructor.

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

- ldapquery.h

6.95 RunPlugin::lib_plugin_t Union Reference

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

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

6.96 gridftpd::RunPlugin::lib_plugin_t Union Reference

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

- gridftpd/run/run_plugin.h

6.97 ARex::LoggerClient Class Reference

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

- client.h

6.98 LRMSResult Class Reference

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

- info_types.h

6.99 ISIS::Neighbor_Container Class Reference

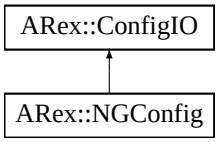
The documentation for this class was generated from the following file:

- isis.h

6.100 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)

6.100.1 Detailed Description

Configuration class used for reading configuration files ARC-style.

6.100.2 Member Function Documentation

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

Read old arc.conf style configuration.

Implements **ARex::ConfigIO** (p. 30).

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

Write configuration to named file.

Implements **ARex::ConfigIO** (p. 30).

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

- ngconfig.h

6.101 ARex2::JobDescription::Notification Class Reference

`#include <job_descr.h>`

6.101.1 Detailed Description

Class represents notification requiest

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

- job_descr.h

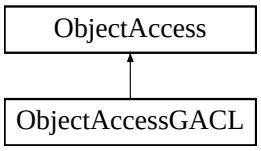
6.102 numvalue_for_shell Class Reference

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

- job_desc.h

6.103 ObjectAccess Class Reference

Inheritance diagram for ObjectAccess:



Data Structures

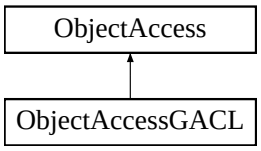
- class **Item**

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

- object_access.h

6.104 ObjectAccessGACL Class Reference

Inheritance diagram for ObjectAccessGACL:



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

- object_access_gacl.h

6.105 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

6.106 ARex::Option Class Reference

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

- **configcore.h**

6.107 ARex2::JobDescription::OutputFile Class Reference

`#include <job_descr.h>`

6.107.1 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

6.108 gridftpd::ParallelLdapQueries Class Reference

`#include <ldapquery.h>`

6.108.1 Detailed Description

General method to perform parallel ldap-queries to a set of clusters

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

- ldapquery.h

6.109 Paul::PaulService Class Reference

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

- paul.h

6.110 ARex::PayloadBigFile Class Reference

Public Member Functions

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

6.110.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.110.1.2 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

6.111 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

6.111.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

6.111.1.2 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

6.112 ARex::PayloadFile Class Reference

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

Public Member Functions

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

6.112.1 Detailed Description

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

6.112.2 Constructor & Destructor Documentation

6.112.2.1 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.

6.112.2.2 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

6.113 Hopi::PayloadFile Class Reference

`#include <PayloadFile.h>`

Public Member Functions

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

6.113.1 Detailed Description

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

6.113.2 Constructor & Destructor Documentation

6.113.2.1 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.

6.113.2.2 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

6.114 DREService::PerlProcessor Class Reference

Data Structures

- struct **ThreadInterface**

Public Member Functions

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

6.114.1 Constructor & Destructor Documentation

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

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

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

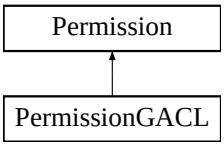
Destructor.

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

- PerlProcessor.h

6.115 Permission Class Reference

Inheritance diagram for Permission:

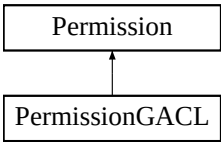


The documentation for this class was generated from the following file:

- permission.h

6.116 PermissionGACL Class Reference

Inheritance diagram for PermissionGACL:



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

- permission_gacl.h

6.117 Policy Class Reference

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

- Policy.h

6.118 ArcSec::Charon::PolicyLocation Class Reference

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

- charon.h

6.119 GridScheduler::Resource Class Reference

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

- resource.h

6.120 GridScheduler::ResourcesHandling Class Reference

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

- resources_handling.h

6.121 ContinuationPlugins::result_t Class Reference

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

- plugins.h

6.122 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

6.122.1 Detailed Description

Interface for handling runtime environments. **RTE** (p. 60) class. It represents a runtime environment, and provides functionality for getting information about them.

6.122.2 Constructor & Destructor Documentation

6.122.2.1 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.

6.122.2.2 RTE::~RTE ( )

Destructor. Not that much to say.

6.122.3 Member Function Documentation

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

Returns the name of the runtime environment.

6.122.3.2 bool RTE::operator!= (const RTE & *other*) const

Inequility operator. Return the opsite of ==

6.122.3.3 bool RTE::operator< (const RTE & *other*) const

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

6.122.3.4 bool RTE::operator<= (const RTE & *other*) const

Less than or equal operator. Returns the oppsite of >

6.122.3.5 bool RTE::operator== (const RTE & *other*) const

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

6.122.3.6 bool RTE::operator> (const RTE & *other*) const

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

6.122.3.7 bool RTE::operator>= (const RTE & *other*) const

Greater or equal operator. Returns the opposite of <

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

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

6.122.3.9 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

6.123 RunFunction Class Reference

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

- run_function.h

6.124 RunParallel Class Reference

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

- run_parallel.h

6.125 gridftpd::RunPlugin Class Reference

Data Structures

- union **lib_plugin_t**

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

- gridftpd/run/run_plugin.h

6.126 RunPlugin Class Reference

Data Structures

- union **lib_plugin_t**

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

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

6.127 RunPlugins Class Reference

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

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

6.128 RunRedirected Class Reference

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

- run_redirected.h

6.129 Server Class Reference

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

- Server.h

6.130 FileRoot::ServerParams Class Reference

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

- fileroot.h

6.131 ArcSec::Service_AA Class Reference

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

6.131.1 Detailed Description

A Service which includes the AttributeAuthority functionality; it accepts the <samlp:AttributeQuery> which includes the <Subject> of the principal from the request and <Attribute> which the request would get; it access some local attribute database and returns <samlp:Assertion> which includes the <Attribute>

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

- aaservice.h

6.132 Compiler::Service_Compiler Class Reference

`#include <compiler.h>`

Public Member Functions

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

6.132.1 Detailed Description

This service need in the server config a "compiler:scriptfile_url" element. It is the scriptfile place, where-from it will be download the JSDL.

6.132.2 Constructor & Destructor Documentation

6.132.2.1 Compiler::Service_Compiler::Service_Compiler (Arc::Config * *cfg*)

Constructor accepts configuration describing content of scriptfile_url

6.132.3 Member Function Documentation

6.132.3.1 virtual Arc::MCC_Status Compiler::Service_Compiler::process (Arc::Message & , Arc::Message &) [virtual]

Service request processing routine

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

- compiler.h

6.133 ArcSec::Service_Delegation Class Reference

A Service which launches the proxy certificate request; it accepts the request from.

`#include <delegation.h>`

6.133.1 Detailed Description

A Service which launches the proxy certificate request; it accepts the request from.

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

- delegation.h

6.134 Arc::Service_JavaWrapper Class Reference

Public Member Functions

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

6.134.1 Member Function Documentation

6.134.1.1 virtual Arc::MCC_Status Arc::Service_JavaWrapper::process (Arc::Message & , Arc::Message &) [virtual]

Service request processing routine

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

- javawrapper.h

6.135 Arc::Service_PythonWrapper Class Reference

Public Member Functions

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

6.135.1 Member Function Documentation

6.135.1.1 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

6.136 ArcSec::Service_SLCS Class Reference

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

6.136.1 Detailed Description

A Service which signs the short-lived certificate; it accepts the certificate signing request (CSR) from from client side through soap, signs a short-lived certificate and sends back through soap. This service is supposed to be deployed together with the SPService and saml2sso.serviceprovider handler, in order to sign certificate based on the authentication result from saml2sso profile. Also the saml attribute (inside the saml assertion from saml2sso profile) will be put into the signed short-lived certificate. By deploying this service together with SPService and saml2sso.serviceprovider handler, we can get the convertion from username/password ------> x509 certificate.

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

- slcs.h

6.137 SPService::Service_SP Class Reference

`#include <SPService.h>`

Public Member Functions

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

6.137.1 Detailed Description

This is service which accepts HTTP request from user agent (web browser) in the client side and processes the functionality of Service Provider in SAML2 SSO profile --- composing <AuthnRequest> Note: the IdP name is provided by the user agent directely when it gives a request, instead of the WRYF(where are you from) or Discovery Service in other implementation

6.137.2 Constructor & Destructor Documentation

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

Constructor

6.137.3 Member Function Documentation

6.137.3.1 virtual Arc::MCC_Status SPService::Service_SP::process (Arc::Message & , Arc::Message &) [virtual]

Service request processing routine

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

- SPService.h

6.138 SimpleMap Class Reference

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

- simplemap.h

6.139 Paul::SysInfo Class Reference

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

- sysinfo.h

6.140 DREService::Task Class Reference

Public Member Functions

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

6.140.1 Constructor & Destructor Documentation

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

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

6.140.1.2 virtual DREService::Task::∼Task (void) [virtual]

Destructor.

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

- Task.h

6.141 DREService::TaskQueue Class Reference

Public Member Functions

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

6.141.1 Constructor & Destructor Documentation

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

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

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

Destructor.

6.141.2 Member Function Documentation

6.141.2.1 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.

6.141.2.2 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

6.142 DREService::TaskSet Class Reference

Public Member Functions

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

6.142.1 Constructor & Destructor Documentation

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

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

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

Destructor.

6.142.2 Member Function Documentation

6.142.2.1 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

6.143 UnixMap Class Reference

Data Structures

- struct **source_t**
- class **unix_user_t**

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

- unixmap.h

6.144 gridftpd::UrlMapConfig Class Reference

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

- gridftpd/conf/conf_map.h

6.145 UrlMapConfig Class Reference

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

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

6.146 userspec_t Class Reference

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

- userspec.h

6.147 value_for_shell Class Reference

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

- job_desc.h

6.148 voms Struct Reference

`#include <auth.h>`

Data Fields

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

6.148.1 Detailed Description

VOMS data

6.148.2 Field Documentation

6.148.2.1 std::string voms::server

The VOMS server DN, as from its certificate

6.148.2.2 std::vector<voms_attrs> voms::std

User's characteristics

6.148.2.3 std::string voms::voname

The name of the VO to which the VOMS belongs

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

- auth.h

6.149 voms_attrs Struct Reference

`#include <auth.h>`

Data Fields

- std::string **group**
- std::string **role**
- std::string **cap**

6.149.1 Detailed Description

VOMS attributes

6.149.2 Field Documentation

6.149.2.1 std::string voms_attrs::cap

user's capability

6.149.2.2 std::string voms_attrs::group

user's group

6.149.2.3 std::string voms_attrs::role

user's role

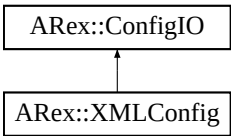
The documentation for this struct was generated from the following file:

- auth.h

6.150 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)

6.150.1 Detailed Description

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

6.150.2 Member Function Documentation

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

Read configuration.

Implements **ARex::ConfigIO** (p. 30).

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

Write configuration.

Implements **ARex::ConfigIO** (p. 30).

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

- xmlconfig.h

6.151 ZeroUInt Class Reference

`#include <states.h>`

6.151.1 Detailed Description

ZeroUInt (p. 71) is a wrapper around unsigned int. It provides a consistent default value, as int type variables have no predefined value assigned upon creation. It also protects from potential counter underflow, to stop counter jumping to MAX_INT.

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

- states.h

Chapter 7

File Documentation

7.1 configcore.h File Reference

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

Data Structures

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

Functions

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

7.1.1 Detailed Description

This file describes the core configuration

Index

~CacheService
  Cache::CacheService, 23
~Compiler_AREXClient
  Arc::Compiler_AREXClient, 25
~DREWebService
  DREService::DREWebService, 32
~Job
  Job, 44
~LdapQuery
  gridftpd::LdapQuery, 51
~PayloadBigFile
  ARex::PayloadBigFile, 56
  Hopi::PayloadBigFile, 56
~PayloadFile
  ARex::PayloadFile, 57
  Hopi::PayloadFile, 57
~PerlProcessor
  DREService::PerlProcessor, 58
~RTE
  RTE, 60
~Task
  DREService::Task, 66
~TaskQueue
  DREService::TaskQueue, 67
~TaskSet
  DREService::TaskSet, 67

Arc::Compiler_AREXClient, 25
  ~Compiler_AREXClient, 25
  clean, 26
  Compiler_AREXClient, 25
  kill, 26
  sstat, 26
  stat, 26
  submit, 27
Arc::Compiler_AREXClientError, 27
  Compiler_AREXClientError, 28
Arc::Compiler_AREXFile, 28
Arc::Job, 43
Arc::JobNotFoundException, 47
Arc::JobQueue, 48
Arc::JobQueueIterator, 48
Arc::JobRequest, 48
Arc::JobSchedMetaData, 48
Arc::JobSelector, 49
Arc::Service_JavaWrapper, 64
  process, 64
Arc::Service_PythonWrapper, 64
  process, 64
ArcSec::Charon, 24
ArcSec::Charon::PolicyLocation, 59
ArcSec::Service_AA, 62
ArcSec::Service_Delegation, 63
ArcSec::Service_SLCS, 64
ARex2::ARex2Service, 17
ARex2::JobControl, 45
ARex2::JobController, 45
ARex2::JobDataCache, 45
ARex2::JobDescription, 46
  JobName, 46
ARex2::JobDescription::InputFile, 40
ARex2::JobDescription::Notification, 53
ARex2::JobDescription::OutputFile, 55
ARex2::JobLRMSInfo, 47
ARex2::JobState, 49
ARex2::JobUser, 50
ARex::ARexGMConfig, 17
ARex::ARexJob, 17
  ARexJob, 18
  Cancel, 18
  ChooseSessionDir, 18
  Clean, 18
  CreateFile, 18
  Failed, 19
  Failure, 19
  GetDescription, 19
  ID, 19
  Jobs, 19
  LogDir, 19
  LogFiles, 19
  OpenDir, 19
  OpenFile, 19
  OpenLogFile, 19
  Resume, 19
  SessionDir, 20
  State, 20
  TotalJobs, 20
  UpdateCredentials, 20
ARex::ARexService, 20
ARex::ConfGrp, 28

ARex::Config, 28
ConfValue, 29
FirstConfValue, 29
GetConfigs, 29
ARex::ConfigError, 29
ConfigError, 29
ARex::ConfigIO, 29
Read, 30
Write, 30
ARex::CountedResource, 31
ARex::FileChunks, 34
Release, 34
Remove, 34
ARex::FileChunksList, 34
Get, 35
GetStuck, 35
ARex::GridManager, 38
ARex::JobRecord, 48
ARex::LoggerClient, 52
ARex::NGConfig, 53
Read, 53
Write, 53
ARex::OptimizedInformationContainer, 55
ARex::Option, 55
ARex::PayloadBigFile, 56
∼PayloadBigFile, 56
PayloadBigFile, 56
ARex::PayloadFile, 56
∼PayloadFile, 57
PayloadFile, 57
ARex::XMLConfig, 70
Read, 70
Write, 70
ARexJob
ARex::ARexJob, 18
AuthEvaluator, 20
AuthUser, 20
AuthVO, 21

Cache::CacheService, 22
∼CacheService, 23
CacheCheck, 23
CacheLink, 23
CacheService, 23
operator bool, 23
process, 24
RegistrationCollector, 24
CacheCheck
Cache::CacheService, 23
CacheConfig, 21
CacheConfig, 21
parseINIConf, 22
setCacheDirs, 22
CacheConfigException, 22
CacheLink
Cache::CacheService, 23
CacheService
Cache::CacheService, 23
Cancel
ARex::ARexJob, 18
Job, 44
cap
voms_attrs, 69
ChooseSessionDir
ARex::ARexJob, 18
Clean
ARex::ARexJob, 18
clean
Arc::Compiler_AREXClient, 26
CommFIFO, 24
Compiler::Service_Compiler, 63
process, 63
Service_Compiler, 63
Compiler_AREXClient
Arc::Compiler_AREXClient, 25
Compiler_AREXClientError
Arc::Compiler_AREXClientError, 28
configcore.h, 73
ConfigError
ARex::ConfigError, 29
ConfigSections, 30
ConfValue
ARex::Config, 29
ContinuationPlugins, 31
ContinuationPlugins::result_t, 59
CreateFile
ARex::ARexJob, 18

deploy
Janitor, 42
DirectAccess, 31
DirectAccess::diraccess_t, 31
DirectFilePlugin, 32
DirEntry, 32
DREService, 15
DREService::DREWebService, 32
∼DREWebService, 32
DREWebService, 32
logger, 33
makeFault, 33
ns_, 33
process, 33
DREService::PerlProcessor, 58
∼PerlProcessor, 58
PerlProcessor, 58
DREService::Task, 66
∼Task, 66
Task, 66

DREService::TaskQueue, 66
~TaskQueue, 67
pushTask, 67
shiftTask, 67
TaskQueue, 67
DREService::TaskSet, 67
~TaskSet, 67
removeTask, 67
TaskSet, 67
DREWebService
DREService::DREWebService, 32

Entry, 34

Failed
ARex::ARexJob, 19
Failure
ARex::ARexJob, 19
FileData, 35
FileNode, 35
FilePlugin, 35
FileRoot, 36
FileRoot::ServerParams, 62
FirstConfValue
ARex::Config, 29
fs_usage, 36

GACLPlugin, 36
Get
ARex::FileChunksList, 35
GetConfigs
ARex::Config, 29
GetDescription
ARex::ARexJob, 19
GetSessionDir
Job, 44
GetState
Job, 44
GetStuck
ARex::FileChunksList, 35
gm_dirs_, 37
GMEnvironment, 37
nordugrid_config_loc, 37
support_mail_address, 37
GridFTP_Commands, 38
GridFTP_Commands_timeout, 38
gridftpd::ConfigSections, 30
gridftpd::Daemon, 31
gridftpd::GMEnvironment, 37
nordugrid_config_loc, 37
support_mail_address, 37
gridftpd::LdapQuery, 50
~LdapQuery, 51
Host, 51
LdapQuery, 51
Query, 51
Result, 51
Scope, 51
gridftpd::LdapQueryError, 51
LdapQueryError, 52
gridftpd::ParallelLdapQueries, 55
gridftpd::RunPlugin, 61
gridftpd::RunPlugin::lib_plugin_t, 52
gridftpd::UrlMapConfig, 68
GridScheduler::GridSchedulerService, 38
GridScheduler::Resource, 59
GridScheduler::ResourcesHandling, 59
group
voms_attrs, 69

Hopi::Hopi, 38
Hopi::PayloadBigFile, 56
~PayloadBigFile, 56
PayloadBigFile, 56
Hopi::PayloadFile, 57
~PayloadFile, 57
PayloadFile, 57
Host
gridftpd::LdapQuery, 51

ID
ARex::ARexJob, 19
Identity, 39
Identity::Item, 41
IdentityGACL, 39
IdentityItemDN, 40
IdentityItemVOMS, 40
Index, 40
ISIS::ISIService, 41
ISIS::ISISSecAttr, 41
ISIS::Neighbor_Container, 53

Janitor, 42
deploy, 42
Janitor, 42
remove, 42
result, 42
wait, 43
Job, 43
~Job, 44
Cancel, 44
GetSessionDir, 44
GetState, 44
Job, 44
operator bool, 44
Resume, 44
Start, 44
job_state_rec_t, 45

JobDescription, 45
JobLocalDescription, 46
JobLog, 46
JobName
    ARex2::JobDescription, 46
JobPlugin, 47
Jobs
    ARex::ARexJob, 19
JobsList, 49
JobsListConfig, 49
JobUser, 49
JobUserHelper, 50
JobUsers, 50

kill
    Arc::Compiler_AREXClient, 26

LdapQuery
    gridftpd::LdapQuery, 51
LdapQueryError
    gridftpd::LdapQueryError, 52
LogDir
    ARex::ARexJob, 19
LogFiles
    ARex::ARexJob, 19
logger
    DREService::DREWebService, 33
LRMSResult, 52

makeFault
    DREService::DREWebService, 33

Name
    RTE, 60
nordugrid_config_loc
    GMEnvironment, 37
    gridftpd::GMEnvironment, 37
ns_
    DREService::DREWebService, 33
numvalue_for_shell, 54

ObjectAccess, 54
ObjectAccess::Item, 41
ObjectAccessGACL, 54
OpenDir
    ARex::ARexJob, 19
OpenFile
    ARex::ARexJob, 19
OpenLogFile
    ARex::ARexJob, 19
operator bool
    Cache::CacheService, 23
    Job, 44
operator<
    RTE, 60
operator<=
    RTE, 60
operator>
    RTE, 61
operator>=
    RTE, 61
operator==
    RTE, 61

parseINIConf
    CacheConfig, 22
Paul::Configurator, 31
Paul::HTMLRequest, 39
Paul::HTMLResponse, 39
Paul::InvalidMessageException, 41
Paul::Job, 43
Paul::JobList, 46
Paul::JobQueue, 47
Paul::JobRequest, 48
Paul::JobSchedMetaData, 48
Paul::PaulService, 55
Paul::SysInfo, 66
PayloadBigFile
    ARex::PayloadBigFile, 56
    Hopi::PayloadBigFile, 56
PayloadFile
    ARex::PayloadFile, 57
    Hopi::PayloadFile, 57
PerlProcessor
    DREService::PerlProcessor, 58
Permission, 58
PermissionGACL, 59
Policy, 59
process
    Arc::Service_JavaWrapper, 64
    Arc::Service_PythonWrapper, 64
    Cache::CacheService, 24
    Compiler::Service_Compiler, 63
    DREService::DREWebService, 33
    SPService::Service_SP, 65
pushTask
    DREService::TaskQueue, 67

Query
    gridftpd::LdapQuery, 51

Read
    ARex::ConfigIO, 30
    ARex::NGConfig, 53
    ARex::XMLConfig, 70
RegistrationCollector
    Cache::CacheService, 24
Release
    ARex::FileChunks, 34

Remove
ARex::FileChunks, 34
remove
Janitor, 42
removeTask
DREService::TaskSet, 67
Result
gridftpd::LdapQuery, 51
result
Janitor, 42
Resume
ARex::ARexJob, 19
Job, 44
role
voms_attrs, 70
RTE, 60
∼RTE, 60
Name, 60
operator<, 60
operator<=, 60
operator>, 61
operator>=, 61
operator==, 61
RTE, 60
str, 61
Version, 61
RunFunction, 61
RunParallel, 61
RunPlugin, 62
RunPlugin::lib_plugin_t, 52
RunPlugins, 62
RunRedirected, 62

Scope
gridftpd::LdapQuery, 51
Server, 62
server
voms, 69
Service_Compiler
Compiler::Service_Compiler, 63
Service_SP
SPService::Service_SP, 65
SessionDir
ARex::ARexJob, 20
setCacheDirs
CacheConfig, 22
shiftTask
DREService::TaskQueue, 67
SimpleMap, 66
SPService::Service_SP, 65
process, 65
Service_SP, 65
sstat
Arc::Compiler_AREXClient, 26
Start
Job, 44
stat
Arc::Compiler_AREXClient, 26
State
ARex::ARexJob, 20
std
voms, 69
str
RTE, 61
submit
Arc::Compiler_AREXClient, 27
support_mail_address
GMEnvironment, 37
gridftpd::GMEnvironment, 37

Task
DREService::Task, 66
TaskQueue
DREService::TaskQueue, 67
TaskSet
DREService::TaskSet, 67
TotalJobs
ARex::ARexJob, 20

UnixMap, 68
UpdateCredentials
ARex::ARexJob, 20
UrlMapConfig, 68
userspec_t, 68

value_for_shell, 68
Version
RTE, 61
voms, 68
server, 69
std, 69
voname, 69
voms_attrs, 69
cap, 69
group, 69
role, 70
voname
voms, 69

wait
Janitor, 43
Write
ARex::ConfigIO, 30
ARex::NGConfig, 53
ARex::XMLConfig, 70

ZeroUInt, 71