

Hosting Environment (Daemon) Services

Generated by Doxygen 1.7.4

Mon Oct 3 2011 14:32:49

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 | Namespace Documentation | 11 |
| 4.1 | DREService Namespace Reference | 11 |
| 4.1.1 | Detailed Description | 11 |
| 5 | Data Structure Documentation | 13 |
| 5.1 | ARex::ARexGMConfig Class Reference | 13 |
| 5.2 | ARex::ARexJob Class Reference | 13 |
| 5.2.1 | Detailed Description | 14 |
| 5.2.2 | Constructor & Destructor Documentation | 14 |
| 5.2.2.1 | ARexJob | 14 |
| 5.2.2.2 | ARexJob | 14 |
| 5.2.3 | Member Function Documentation | 14 |
| 5.2.3.1 | Cancel | 14 |
| 5.2.3.2 | ChooseSessionDir | 14 |
| 5.2.3.3 | Clean | 15 |
| 5.2.3.4 | CreateFile | 15 |
| 5.2.3.5 | Failed | 15 |
| 5.2.3.6 | Failure | 15 |

| | | |
|----------|--|----|
| 5.2.3.7 | GetDescription | 15 |
| 5.2.3.8 | ID | 15 |
| 5.2.3.9 | Jobs | 15 |
| 5.2.3.10 | LogDir | 15 |
| 5.2.3.11 | LogFiles | 15 |
| 5.2.3.12 | OpenDir | 15 |
| 5.2.3.13 | OpenFile | 16 |
| 5.2.3.14 | OpenLogFile | 16 |
| 5.2.3.15 | Resume | 16 |
| 5.2.3.16 | SessionDir | 16 |
| 5.2.3.17 | State | 16 |
| 5.2.3.18 | State | 16 |
| 5.2.3.19 | TotalJobs | 16 |
| 5.2.3.20 | UpdateCredentials | 16 |
| 5.3 | ARex::ARexService Class Reference | 16 |
| 5.4 | AuthEvaluator Class Reference | 17 |
| 5.5 | AuthUser Class Reference | 17 |
| 5.6 | AuthVO Class Reference | 17 |
| 5.7 | CacheConfig Class Reference | 17 |
| 5.7.1 | Detailed Description | 18 |
| 5.7.2 | Constructor & Destructor Documentation | 18 |
| 5.7.2.1 | CacheConfig | 18 |
| 5.7.2.2 | CacheConfig | 18 |
| 5.7.3 | Member Function Documentation | 18 |
| 5.7.3.1 | parseINIConf | 18 |
| 5.7.3.2 | setCacheDirs | 18 |
| 5.8 | CacheConfigException Class Reference | 18 |
| 5.8.1 | Detailed Description | 18 |
| 5.9 | Cache::CacheService Class Reference | 19 |
| 5.9.1 | Detailed Description | 19 |
| 5.9.2 | Constructor & Destructor Documentation | 19 |
| 5.9.2.1 | CacheService | 19 |
| 5.9.2.2 | ~CacheService | 19 |
| 5.9.3 | Member Function Documentation | 20 |

| | | |
|----------|--|----|
| 5.9.3.1 | CacheCheck | 20 |
| 5.9.3.2 | CacheLink | 20 |
| 5.9.3.3 | operator bool | 20 |
| 5.9.3.4 | operator! | 20 |
| 5.9.3.5 | process | 20 |
| 5.9.3.6 | RegistrationCollector | 20 |
| 5.10 | ArcSec::Charon Class Reference | 21 |
| 5.10.1 | Detailed Description | 21 |
| 5.11 | CommFIFO Class Reference | 21 |
| 5.12 | ConfigSections Class Reference | 21 |
| 5.13 | gridftpdc::ConfigSections Class Reference | 21 |
| 5.14 | ContinuationPlugins Class Reference | 22 |
| 5.15 | ARex::CountedResource Class Reference | 22 |
| 5.16 | gridftpdc::Daemon Class Reference | 22 |
| 5.17 | DataStaging::DataDeliveryService Class Reference | 22 |
| 5.17.1 | Detailed Description | 23 |
| 5.18 | DirectAccess::diraccess_t Struct Reference | 23 |
| 5.19 | DirectAccess Class Reference | 23 |
| 5.20 | DirectFilePlugin Class Reference | 24 |
| 5.21 | DirEntry Class Reference | 24 |
| 5.22 | DREService::DREWebService Class Reference | 24 |
| 5.22.1 | Constructor & Destructor Documentation | 25 |
| 5.22.1.1 | DREWebService | 25 |
| 5.22.1.2 | ~DREWebService | 25 |
| 5.22.2 | Member Function Documentation | 25 |
| 5.22.2.1 | makeFault | 25 |
| 5.22.2.2 | process | 25 |
| 5.22.3 | Field Documentation | 25 |
| 5.22.3.1 | logger | 25 |
| 5.22.3.2 | ns_ | 26 |
| 5.23 | DTRGenerator Class Reference | 26 |
| 5.23.1 | Detailed Description | 26 |
| 5.23.2 | Constructor & Destructor Documentation | 26 |
| 5.23.2.1 | DTRGenerator | 26 |

| | | |
|----------|--|----|
| 5.23.2.2 | ~DTRGenerator | 26 |
| 5.23.3 | Member Function Documentation | 27 |
| 5.23.3.1 | cancelJob | 27 |
| 5.23.3.2 | checkUploadedFiles | 27 |
| 5.23.3.3 | queryJobFinished | 27 |
| 5.23.3.4 | receiveDTR | 27 |
| 5.23.3.5 | receiveJob | 28 |
| 5.24 | DTRInfo Class Reference | 28 |
| 5.24.1 | Detailed Description | 28 |
| 5.24.2 | Constructor & Destructor Documentation | 28 |
| 5.24.2.1 | DTRInfo | 28 |
| 5.25 | Entry Class Reference | 28 |
| 5.26 | ARex::FileChunks Class Reference | 29 |
| 5.26.1 | Detailed Description | 29 |
| 5.26.2 | Member Function Documentation | 29 |
| 5.26.2.1 | Release | 29 |
| 5.26.2.2 | Remove | 29 |
| 5.27 | ARex::FileChunksList Class Reference | 29 |
| 5.27.1 | Detailed Description | 30 |
| 5.27.2 | Member Function Documentation | 30 |
| 5.27.2.1 | Get | 30 |
| 5.27.2.2 | GetStuck | 30 |
| 5.28 | FileData Class Reference | 30 |
| 5.29 | FileNode Class Reference | 30 |
| 5.30 | FilePlugin Class Reference | 31 |
| 5.31 | FileRoot Class Reference | 31 |
| 5.32 | GACLPlugin Class Reference | 31 |
| 5.33 | gm_dirs_ Struct Reference | 32 |
| 5.34 | GMEEnvironment Class Reference | 32 |
| 5.34.1 | Member Function Documentation | 32 |
| 5.34.1.1 | nordugrid_config_loc | 32 |
| 5.34.1.2 | support_mail_address | 32 |
| 5.35 | gridftpd::GMEEnvironment Class Reference | 32 |
| 5.35.1 | Member Function Documentation | 33 |

| | | |
|----------|--|----|
| 5.35.1.1 | nordugrid_config_loc | 33 |
| 5.35.1.2 | support_mail_address | 33 |
| 5.36 | GridFTP_Commands Class Reference | 33 |
| 5.37 | GridFTP_Commands_timeout Class Reference | 33 |
| 5.38 | ARex::GridManager Class Reference | 33 |
| 5.39 | Hopi::Hopi Class Reference | 34 |
| 5.40 | Identity Class Reference | 34 |
| 5.41 | IdentityGACL Class Reference | 34 |
| 5.42 | IdentityItemDN Class Reference | 34 |
| 5.43 | IdentityItemVOMS Class Reference | 35 |
| 5.44 | Index Class Reference | 35 |
| 5.45 | ISIS::ISIService Class Reference | 35 |
| 5.46 | ISIS::ISISecAttr Class Reference | 35 |
| 5.47 | ObjectAccess::Item Class Reference | 36 |
| 5.48 | Identity::Item Class Reference | 36 |
| 5.49 | Janitor Class Reference | 36 |
| 5.49.1 | Detailed Description | 37 |
| 5.49.2 | Constructor & Destructor Documentation | 37 |
| 5.49.2.1 | Janitor | 37 |
| 5.49.3 | Member Function Documentation | 37 |
| 5.49.3.1 | deploy | 37 |
| 5.49.3.2 | remove | 37 |
| 5.49.3.3 | result | 37 |
| 5.49.3.4 | wait | 37 |
| 5.50 | job_state_rec_t Struct Reference | 38 |
| 5.51 | JobDescription Class Reference | 38 |
| 5.52 | JobLocalDescription Class Reference | 38 |
| 5.53 | JobLog Class Reference | 38 |
| 5.53.1 | Detailed Description | 38 |
| 5.54 | JobPlugin Class Reference | 38 |
| 5.55 | ARex::JobRecord Class Reference | 39 |
| 5.56 | JobsList Class Reference | 39 |
| 5.57 | JobsListConfig Class Reference | 39 |
| 5.57.1 | Detailed Description | 39 |

| | | |
|----------|---|----|
| 5.58 | JobUser Class Reference | 39 |
| 5.59 | JobUserHelper Class Reference | 40 |
| 5.60 | JobUsers Class Reference | 40 |
| 5.61 | gridftp::LdapQuery Class Reference | 40 |
| 5.61.1 | Detailed Description | 40 |
| 5.61.2 | Member Enumeration Documentation | 40 |
| 5.61.2.1 | Scope | 40 |
| 5.61.3 | Constructor & Destructor Documentation | 41 |
| 5.61.3.1 | LdapQuery | 41 |
| 5.61.3.2 | ~LdapQuery | 41 |
| 5.61.4 | Member Function Documentation | 41 |
| 5.61.4.1 | Host | 41 |
| 5.61.4.2 | Query | 41 |
| 5.61.4.3 | Result | 41 |
| 5.62 | gridftp::LdapQueryError Class Reference | 41 |
| 5.62.1 | Detailed Description | 42 |
| 5.62.2 | Constructor & Destructor Documentation | 42 |
| 5.62.2.1 | LdapQueryError | 42 |
| 5.63 | RunPlugin::lib_plugin_t Union Reference | 42 |
| 5.64 | gridftp::RunPlugin::lib_plugin_t Union Reference | 42 |
| 5.65 | ARex::LoggerClient Class Reference | 42 |
| 5.66 | LRMSResult Class Reference | 42 |
| 5.67 | ISIS::Neighbor_Container Class Reference | 42 |
| 5.68 | numvalue_for_shell Class Reference | 43 |
| 5.69 | ObjectAccess Class Reference | 43 |
| 5.70 | ObjectAccessGACL Class Reference | 43 |
| 5.71 | ARex::OptimizedInformationContainer Class Reference | 44 |
| 5.72 | gridftp::ParallelLdapQueries Class Reference | 44 |
| 5.72.1 | Detailed Description | 44 |
| 5.73 | ARex::PayloadBigFile Class Reference | 44 |
| 5.73.1 | Constructor & Destructor Documentation | 44 |
| 5.73.1.1 | PayloadBigFile | 44 |
| 5.73.1.2 | ~PayloadBigFile | 44 |
| 5.74 | Hopi::PayloadBigFile Class Reference | 45 |

| | | |
|----------|--|----|
| 5.74.1 | Constructor & Destructor Documentation | 45 |
| 5.74.1.1 | PayloadBigFile | 45 |
| 5.74.1.2 | ~PayloadBigFile | 45 |
| 5.75 | ARex::PayloadFAFile Class Reference | 45 |
| 5.75.1 | Constructor & Destructor Documentation | 45 |
| 5.75.1.1 | PayloadFAFile | 45 |
| 5.76 | Hopi::PayloadFile Class Reference | 45 |
| 5.76.1 | Detailed Description | 46 |
| 5.76.2 | Constructor & Destructor Documentation | 46 |
| 5.76.2.1 | PayloadFile | 46 |
| 5.76.2.2 | ~PayloadFile | 46 |
| 5.77 | ARex::PayloadFile Class Reference | 46 |
| 5.77.1 | Detailed Description | 46 |
| 5.77.2 | Constructor & Destructor Documentation | 46 |
| 5.77.2.1 | PayloadFile | 46 |
| 5.77.2.2 | ~PayloadFile | 47 |
| 5.78 | DREService::PerlProcessor Class Reference | 47 |
| 5.78.1 | Constructor & Destructor Documentation | 47 |
| 5.78.1.1 | PerlProcessor | 47 |
| 5.78.1.2 | ~PerlProcessor | 47 |
| 5.79 | Permission Class Reference | 47 |
| 5.80 | PermissionGACL Class Reference | 48 |
| 5.81 | Policy Class Reference | 48 |
| 5.82 | ArcSec::Charon::PolicyLocation Class Reference | 48 |
| 5.83 | ContinuationPlugins::result_t Class Reference | 48 |
| 5.84 | RunFunction Class Reference | 48 |
| 5.85 | RunParallel Class Reference | 49 |
| 5.86 | RunPlugin Class Reference | 49 |
| 5.87 | gridftp::RunPlugin Class Reference | 49 |
| 5.88 | RunPlugins Class Reference | 49 |
| 5.89 | RunRedirected Class Reference | 49 |
| 5.90 | Server Class Reference | 50 |
| 5.91 | FileRoot::ServerParams Class Reference | 50 |
| 5.92 | ArcSec::Service_AA Class Reference | 50 |

| | |
|---|----|
| 5.92.1 Detailed Description | 50 |
| 5.93 Arc::Service_JavaWrapper Class Reference | 50 |
| 5.93.1 Member Function Documentation | 50 |
| 5.93.1.1 process | 50 |
| 5.94 Arc::Service_PythonWrapper Class Reference | 51 |
| 5.94.1 Member Function Documentation | 51 |
| 5.94.1.1 process | 51 |
| 5.95 ArcSec::Service_SLCS Class Reference | 51 |
| 5.95.1 Detailed Description | 51 |
| 5.96 SPService::Service_SP Class Reference | 51 |
| 5.96.1 Detailed Description | 52 |
| 5.96.2 Constructor & Destructor Documentation | 52 |
| 5.96.2.1 Service_SP | 52 |
| 5.96.3 Member Function Documentation | 52 |
| 5.96.3.1 process | 52 |
| 5.97 SimpleMap Class Reference | 52 |
| 5.98 DREService::Task Class Reference | 52 |
| 5.98.1 Constructor & Destructor Documentation | 52 |
| 5.98.1.1 Task | 52 |
| 5.98.1.2 ~Task | 53 |
| 5.99 DREService::TaskQueue Class Reference | 53 |
| 5.99.1 Constructor & Destructor Documentation | 53 |
| 5.99.1.1 TaskQueue | 53 |
| 5.99.1.2 ~TaskQueue | 53 |
| 5.99.2 Member Function Documentation | 53 |
| 5.99.2.1 pushTask | 53 |
| 5.99.2.2 shiftTask | 53 |
| 5.100 DREService::TaskSet Class Reference | 54 |
| 5.100.1 Constructor & Destructor Documentation | 54 |
| 5.100.1.1 TaskSet | 54 |
| 5.100.1.2 ~TaskSet | 54 |
| 5.100.2 Member Function Documentation | 54 |
| 5.100.2.1 removeTask | 54 |
| 5.101 UnixMap Class Reference | 54 |

| | |
|--|----|
| 5.102gridftp::UrlMapConfig Class Reference | 55 |
| 5.103UrlMapConfig Class Reference | 55 |
| 5.104userspec_t Class Reference | 55 |
| 5.105value_for_shell Class Reference | 55 |
| 5.106voms Struct Reference | 55 |
| 5.106.1 Detailed Description | 55 |
| 5.106.2 Field Documentation | 56 |
| 5.106.2.1 attrs | 56 |
| 5.106.2.2 server | 56 |
| 5.106.2.3 voname | 56 |
| 5.107voms_attrs Struct Reference | 56 |
| 5.107.1 Detailed Description | 56 |
| 5.107.2 Field Documentation | 56 |
| 5.107.2.1 cap | 56 |
| 5.107.2.2 group | 56 |
| 5.107.2.3 role | 57 |
| 5.108ZeroUInt Class Reference | 57 |
| 5.108.1 Detailed Description | 57 |

Chapter 1

Namespace Index

1.1 Namespace List

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

| | |
|--------------------------------------|----|
| DREService | 11 |
|--------------------------------------|----|

Chapter 2

Data Structure Index

2.1 Class Hierarchy

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

| | |
|--|----|
| ARex::ARexGMConfig | 13 |
| ARex::ARexJob | 13 |
| ARex::ARexService | 16 |
| AuthEvaluator | 17 |
| AuthUser | 17 |
| AuthVO | 17 |
| CacheConfig | 17 |
| CacheConfigException | 18 |
| Cache::CacheService | 19 |
| ArcSec::Charon | 21 |
| CommFIFO | 21 |
| ConfigSections | 21 |
| gridftpd::ConfigSections | 21 |
| ContinuationPlugins | 22 |
| ARex::CountedResource | 22 |
| gridftpd::Daemon | 22 |
| DataStaging::DataDeliveryService | 22 |
| DirectAccess::diraccess_t | 23 |
| DirectAccess | 23 |
| DirEntry | 24 |
| DREService::DREWebService | 24 |
| DTRGenerator | 26 |
| DTRInfo | 28 |
| Entry | 28 |
| ARex::FileChunks | 29 |
| ARex::FileChunksList | 29 |
| FileData | 30 |
| FileNode | 30 |
| FilePlugin | 31 |

| | |
|---|----|
| DirectFilePlugin | 24 |
| GACLPlugin | 31 |
| JobPlugin | 38 |
| FileRoot | 31 |
| gm_dirs_ | 32 |
| GMEEnvironment | 32 |
| gridftpd::GMEEnvironment | 32 |
| GridFTP_Commands | 33 |
| GridFTP_Commands_timeout | 33 |
| ARex::GridManager | 33 |
| Hopi::Hopi | 34 |
| Identity | 34 |
| IdentityGACL | 34 |
| Index | 35 |
| ISIS::ISIService | 35 |
| ISIS::ISISecAttr | 35 |
| Identity::Item | 36 |
| IdentityItemDN | 34 |
| IdentityItemVOMS | 35 |
| ObjectAccess::Item | 36 |
| Janitor | 36 |
| job_state_rec_t | 38 |
| JobDescription | 38 |
| JobLocalDescription | 38 |
| JobLog | 38 |
| ARex::JobRecord | 39 |
| JobsList | 39 |
| JobsListConfig | 39 |
| JobUser | 39 |
| JobUserHelper | 40 |
| JobUsers | 40 |
| gridftpd::LdapQuery | 40 |
| gridftpd::LdapQueryError | 41 |
| RunPlugin::lib_plugin_t | 42 |
| gridftpd::RunPlugin::lib_plugin_t | 42 |
| ARex::LoggerClient | 42 |
| LRMSResult | 42 |
| ISIS::Neighbor_Container | 42 |
| numvalue_for_shell | 43 |
| ObjectAccess | 43 |
| ObjectAccessGACL | 43 |
| ARex::OptimizedInformationContainer | 44 |
| gridftpd::ParallelLdapQueries | 44 |
| ARex::PayloadBigFile | 44 |
| Hopi::PayloadBigFile | 45 |
| ARex::PayloadFAFile | 45 |
| Hopi::PayloadFile | 45 |
| ARex::PayloadFile | 46 |
| DREService::PerlProcessor | 47 |

| | |
|--|----|
| Permission | 47 |
| PermissionGACL | 48 |
| Policy | 48 |
| ArcSec::Charon::PolicyLocation | 48 |
| ContinuationPlugins::result_t | 48 |
| RunFunction | 48 |
| RunParallel | 49 |
| RunPlugin | 49 |
| gridftpd::RunPlugin | 49 |
| RunPlugins | 49 |
| RunRedirected | 49 |
| Server | 50 |
| FileRoot::ServerParams | 50 |
| ArcSec::Service_AA | 50 |
| Arc::Service_JavaWrapper | 50 |
| Arc::Service_PythonWrapper | 51 |
| ArcSec::Service_SLCS | 51 |
| SPService::Service_SP | 51 |
| SimpleMap | 52 |
| DREService::Task | 52 |
| DREService::TaskQueue | 53 |
| DREService::TaskSet | 54 |
| UnixMap | 54 |
| gridftpd::UrlMapConfig | 55 |
| UrlMapConfig | 55 |
| userspec_t | 55 |
| value_for_shell | 55 |
| voms | 55 |
| voms_attrs | 56 |
| ZeroUInt | 57 |

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

| | |
|--|----|
| ARex::ARexGMConfig | 13 |
| ARex::ARexJob | 13 |
| ARex::ARexService | 16 |
| AuthEvaluator | 17 |
| AuthUser | 17 |
| AuthVO | 17 |
| CacheConfig | 17 |
| CacheConfigException | 18 |
| Cache::CacheService | 19 |
| ArcSec::Charon | 21 |
| CommFIFO | 21 |
| ConfigSections | 21 |
| gridftpd::ConfigSections | 21 |
| ContinuationPlugins | 22 |
| ARex::CountedResource | 22 |
| gridftpd::Daemon | 22 |
| DataStaging::DataDeliveryService (Service for the Delivery layer of data staging) | 22 |
| DirectAccess::diraccess_t | 23 |
| DirectAccess | 23 |
| DirectFilePlugin | 24 |
| DirEntry | 24 |
| DREService::DREWebService | 24 |
| DTRGenerator | 26 |
| DTRInfo | 28 |
| Entry | 28 |
| ARex::FileChunks (Representation of delivered file chunks) | 29 |
| ARex::FileChunksList (Container for FileChunks instances) | 29 |
| FileData | 30 |

| | |
|--|----|
| FileNode | 30 |
| FilePlugin | 31 |
| FileRoot | 31 |
| GACLPlugin | 31 |
| gm_dirs_ | 32 |
| GMEnvironment | 32 |
| gridftpd::GMEnvironment | 32 |
| GridFTP_Commands | 33 |
| GridFTP_Commands_timeout | 33 |
| ARex::GridManager | 33 |
| Hopi::Hopi | 34 |
| Identity | 34 |
| IdentityGACL | 34 |
| IdentityItemDN | 34 |
| IdentityItemVOMS | 35 |
| Index | 35 |
| ISIS::ISIService | 35 |
| ISIS::ISISecAttr | 35 |
| ObjectAccess::Item | 36 |
| Identity::Item | 36 |
| Janitor (Class to communicate with Janitor - Dynmaic Runtime Environment handler) | 36 |
| job_state_rec_t | 38 |
| JobDescription | 38 |
| JobLocalDescription | 38 |
| JobLog | 38 |
| JobPlugin | 38 |
| ARex::JobRecord | 39 |
| JobsList | 39 |
| JobsListConfig | 39 |
| JobUser | 39 |
| JobUserHelper | 40 |
| JobUsers | 40 |
| gridftpd::LdapQuery | 40 |
| gridftpd::LdapQueryError | 41 |
| RunPlugin::lib_plugin_t | 42 |
| gridftpd::RunPlugin::lib_plugin_t | 42 |
| ARex::LoggerClient | 42 |
| LRMSResult | 42 |
| ISIS::Neighbor_Container | 42 |
| numvalue_for_shell | 43 |
| ObjectAccess | 43 |
| ObjectAccessGACL | 43 |
| ARex::OptimizedInformationContainer | 44 |
| gridftpd::ParallelLdapQueries | 44 |
| ARex::PayloadBigFile | 44 |
| Hopi::PayloadBigFile | 45 |
| ARex::PayloadFAFile | 45 |
| Hopi::PayloadFile | 45 |
| ARex::PayloadFile | 46 |

| | |
|--------------------------------|----|
| DREService::PerlProcessor | 47 |
| Permission | 47 |
| PermissionGACL | 48 |
| Policy | 48 |
| ArcSec::Charon::PolicyLocation | 48 |
| ContinuationPlugins::result_t | 48 |
| RunFunction | 48 |
| RunParallel | 49 |
| RunPlugin | 49 |
| gridftpd::RunPlugin | 49 |
| RunPlugins | 49 |
| RunRedirected | 49 |
| Server | 50 |
| FileRoot::ServerParams | 50 |
| ArcSec::Service_AA | 50 |
| Arc::Service_JavaWrapper | 50 |
| Arc::Service_PythonWrapper | 51 |
| ArcSec::Service_SLCS | 51 |
| SPService::Service_SP | 51 |
| SimpleMap | 52 |
| DREService::Task | 52 |
| DREService::TaskQueue | 53 |
| DREService::TaskSet | 54 |
| UnixMap | 54 |
| gridftpd::UrlMapConfig | 55 |
| UrlMapConfig | 55 |
| userspec_t | 55 |
| value_for_shell | 55 |
| voms | 55 |
| voms_attrs | 56 |
| ZeroUInt | 57 |

Chapter 4

Namespace Documentation

4.1 DREService Namespace Reference

Data Structures

- class [DREWebService](#)
- class [PerlProcessor](#)
- class [Task](#)
- class [TaskQueue](#)
- class [TaskSet](#)

4.1.1 Detailed Description

Implementation of a simple echo service

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

Chapter 5

Data Structure Documentation

5.1 ARex::ARexGMConfig Class Reference

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

- job.h

5.2 ARex::ARexJob Class Reference

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

Public Member Functions

- [ARexJob](#) (const std::string &id, [ARexGMConfig](#) &config, Arc::Logger &logger, bool fast_auth_check=false)
- [ARexJob](#) (Arc::XMLNode jsdl, [ARexGMConfig](#) &config, const std::string &credentials, const std::string &clientid, Arc::Logger &logger, Arc::XMLNode migration=Arc::XMLNode())
- std::string [Failure](#) (void)
- std::string [ID](#) (void)
- bool [GetDescription](#) (Arc::XMLNode &jsdl)
- bool [Cancel](#) (void)
- bool [Clean](#) (void)
- bool [Resume](#) (void)
- std::string [State](#) (void)
- std::string [State](#) (bool &job_pending)
- bool [Failed](#) (void)
- std::string [SessionDir](#) (void)
- std::string [LogDir](#) (void)
- Arc::FileAccess * [CreateFile](#) (const std::string &filename)

- `Arc::FileAccess * OpenFile (const std::string &filename, bool for_read, bool for_write)`
- `int OpenLogFile (const std::string &name)`
- `Arc::FileAccess * 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)`

5.2.1 Detailed Description

This class represents convenience interface to manage jobs handled by Grid Manager. It works mostly through corresponding classes and functions of Grid Manager.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `ARex::ARexJob::ARexJob (const std::string & id, ARexGMConfig & config, Arc::Logger & logger, bool fast_auth_check = false)`

Create instance which is an interface to existing job

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

Create new job with provided JSDL description

5.2.3 Member Function Documentation

5.2.3.1 `bool ARex::ARexJob::Cancel (void)`

Cancel processing/execution of job

5.2.3.2 `bool ARex::ARexJob::ChooseSessionDir (const std::string & jobid, std::string & sessiondir)`

Select a session dir to use for this job

5.2.3.3 `bool ARex::ARexJob::Clean (void)`

Remove job from local pool

5.2.3.4 `Arc::FileAccess* ARex::ARexJob::CreateFile (const std::string & filename)`

Creates file in job's session directory and returns handler

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

Returns true if job has failed

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

Returns textual description of failure of last operation

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

Fills provided jsdl with job description

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

Return ID assigned to job

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

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

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

Returns name of virtual log directory

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

Returns list of existing log files

5.2.3.12 `Arc::FileAccess* ARex::ARexJob::OpenDir (const std::string & dirname)`

Opens directory inside session directory

5.2.3.13 `Arc::FileAccess* ARex::ARexJob::OpenFile (const std::string & filename, bool for_read, bool for_write)`

Opens file in job's session directory and returns handler

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

Opens log file in control directory

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

Resume execution of job after error

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

Returns path to session directory

5.2.3.17 `std::string ARex::ARexJob::State (bool & job_pending)`

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

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

Returns current state of job

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

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

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

Updates job credentials

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

- `job.h`

5.3 ARex::ARexService Class Reference

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

- [arex.h](#)

5.4 AuthEvaluator Class Reference

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

- [auth.h](#)

5.5 AuthUser Class Reference

Data Structures

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

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

- [auth.h](#)

5.6 AuthVO Class Reference

Friends

- class [AuthUser](#)

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

- [auth.h](#)

5.7 CacheConfig Class Reference

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

Public Member Functions

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

5.7.1 Detailed Description

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

5.7.2 Constructor & Destructor Documentation

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

Create a new [CacheConfig](#) 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.

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

Empty [CacheConfig](#)

5.7.3 Member Function Documentation

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

Parsers for the two different conf styles

5.7.3.2 `void CacheConfig::setCacheDirs (std::vector< std::string > cache_dirs) [inline]`

To allow for substitutions done during configuration

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

- `conf_cache.h`

5.8 CacheConfigException Class Reference

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

5.8.1 Detailed Description

Exception thrown by constructor caused by bad cache params in conf file

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

- `conf_cache.h`

5.9 Cache::CacheService Class Reference

```
#include <CacheService.h>
```

Public Member Functions

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

Protected Member Functions

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

5.9.1 Detailed Description

[CacheService](#) 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.

5.9.2 Constructor & Destructor Documentation

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

Make a new [CacheService](#). Reads the configuration and determines the validity of the service.

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

Destroy the [CacheService](#)

5.9.3 Member Function Documentation

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

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

Parameters

| | |
|-------------|--|
| <i>user</i> | A-REX user configuration for the mapped user |
|-------------|--|

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

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

Parameters

| | |
|--------------------|--|
| <i>user</i> | A-REX user configuration for the mapped user |
| <i>mapped_user</i> | The local user to which the client DN was mapped |

5.9.3.3 `Cache::CacheService::operator bool (void)` `[inline]`

Returns true if the [CacheService](#) is valid.

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

Returns true if the [CacheService](#) is not valid.

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

Main method called by HED when [CacheService](#) is invoked. Directs call to appropriate [CacheService](#) method.

5.9.3.6 `bool Cache::CacheService::RegistrationCollector (Arc::XMLNode & doc)`

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

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

- [CacheService.h](#)

5.10 ArcSec::Charon Class Reference

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

Data Structures

- class [PolicyLocation](#)

5.10.1 Detailed Description

A Service which includes the ArcPDP functionality; it can be deployed as an independent service to provide request evaluation functionality for the other remote services

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

- [charon.h](#)

5.11 CommFIFO Class Reference

Data Structures

- class [elem_t](#)

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

- [commfifo.h](#)

5.12 ConfigSections Class Reference

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

- [a-rex/grid-manager/conf/conf_sections.h](#)

5.13 gridftpd::ConfigSections Class Reference

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

- [gridftpd/conf/conf_sections.h](#)

5.14 ContinuationPlugins Class Reference

Data Structures

- class **command_t**
- class [result_t](#)

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

- plugins.h

5.15 ARex::CountedResource Class Reference

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

- arex.h

5.16 gridftpd::Daemon Class Reference

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

- daemon.h

5.17 DataStaging::DataDeliveryService Class Reference

Service for the Delivery layer of data staging.

```
#include <DataDeliveryService.h>
```

Public Member Functions

- [DataDeliveryService](#) (Arc::Config *cfg)
- virtual [~DataDeliveryService](#) ()
- virtual Arc::MCC_Status [process](#) (Arc::Message &inmsg, Arc::Message &outmsg)
- virtual void [receiveDTR](#) (DTR &dtr)
- bool [RegistrationCollector](#) (Arc::XMLNode &doc)
- [operator bool](#) () const
- bool [operator!](#) () const

5.17.1 Detailed Description

Service for the Delivery layer of data staging.

This service starts and controls data transfers. It assumes that the files in any request submitted are ready for immediate transfer and so do not need to be resolved or prepared in any way.

It implements DTRCallback to get callbacks when a DTR has finished transfer.

Status codes in results returned:

- OK - successful submission/cancellation
- TRANSFERRING - transfer still ongoing
- TRANSFERRED - transfer finished successfully
- TRANSFER_ERROR - transfer failed
- SERVICE_ERROR - something went wrong in the service itself

An internal list of active transfers is held in memory. After the first query of a finished transfer (successful or not) the DTR is moved to an archived list where only summary information is kept about the transfer (DTR ID, state and short error description). The DTR object is then deleted. This archived list is also kept in memory. In case a transfer is never queried, a separate thread moves any transfers which completed more than one hour ago to the archived list.

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

- DataDeliveryService.h

5.18 DirectAccess::diraccess_t Struct Reference

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

- fileplugin.h

5.19 DirectAccess Class Reference

Data Structures

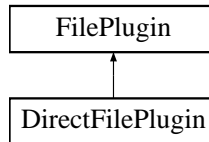
- struct [diraccess_t](#)

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

- fileplugin.h

5.20 DirectFilePlugin Class Reference

Inheritance diagram for DirectFilePlugin:



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

- fileplugin.h

5.21 DirEntry Class Reference

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

- fileroot.h

5.22 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](#)

5.22.1 Constructor & Destructor Documentation

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

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

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

Destructor.

5.22.2 Member Function Documentation

5.22.2.1 Arc::MCC_Status DREService::DREWebService::makeFault (Arc::Message & *outmsg*, const std::string & *reason*) [protected]

Method to return an error. Creates a fault message and returns a status.

Parameters

| | |
|---------------|------------------|
| <i>outmsg</i> | outgoing message |
|---------------|------------------|

Returns

Status of the result achieved

5.22.2.2 virtual Arc::MCC_Status DREService::DREWebService::process (Arc::Message & *inmsg*, Arc::Message & *outmsg*) [virtual]

Implementation of the virtual method defined in MCCInterface (to be found in MCC.h).

Parameters

| | |
|--------------|------------------|
| <i>inmsg</i> | incoming message |
| <i>inmsg</i> | outgoing message |

Returns

Status of the result achieved

5.22.3 Field Documentation

5.22.3.1 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.

5.22.3.2 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

5.23 DTRGenerator Class Reference

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

Public Member Functions

- [DTRGenerator](#) (const [JobUsers](#) &users, void(*kicker_func)(void *)=NULL, void *kicker_arg=NULL)
- [~DTRGenerator](#) ()
- virtual void [receiveDTR](#) (DataStaging::DTR &dtr)
- void [receiveJob](#) (const [JobDescription](#) &job)
- void [cancelJob](#) (const [JobDescription](#) &job)
- bool [queryJobFinished](#) ([JobDescription](#) &job)
- int [checkUploadedFiles](#) ([JobDescription](#) &job)

5.23.1 Detailed Description

A-REX implementation of DTR Generator.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 DTRGenerator::DTRGenerator (const [JobUsers](#) & users, void(*) (void *) *kicker_func* = NULL, void * *kicker_arg* = NULL)

Start up Generator.

Parameters

| | |
|--------------------|--|
| <i>user</i> | JobUsers for this Generator. |
| <i>kicker_func</i> | Function to call on completion of all DTRs for a job |
| <i>kicker_arg</i> | Argument to kicker function |

5.23.2.2 DTRGenerator::~DTRGenerator ()

Stop Generator

5.23.3 Member Function Documentation

5.23.3.1 void DTRGenerator::cancelJob (const JobDescription & job)

This method is used by A-REX to cancel on-going DTRs. A cancel request is made for each DTR in the job and the method returns. The Scheduler asynchronously deals with cancelling the DTRs.

Parameters

| | |
|------------|----------------------------------|
| <i>job</i> | The job which is being cancelled |
|------------|----------------------------------|

5.23.3.2 int DTRGenerator::checkUploadedFiles (JobDescription & job)

Utility method to check that all files the user was supposed to upload with the job are ready.

Parameters

| | |
|------------|---|
| <i>job</i> | Job description, failures will be reported directly in this object. |
|------------|---|

Returns

0 if file exists, 1 if it is not a proper file or other error, 2 if the file not there yet

5.23.3.3 bool DTRGenerator::queryJobFinished (JobDescription & job)

Query status of DTRs in job. If all DTRs are finished, returns true, otherwise returns false. If true is returned, the [JobDescription](#) should be checked for whether the staging was successful or not by checking `GetFailure()`.

Parameters

| | |
|------------|---|
| <i>job</i> | Description of job to query. Can be modified to add a failure reason. |
|------------|---|

Returns

True if all DTRs in the job are finished, false otherwise.

5.23.3.4 virtual void DTRGenerator::receiveDTR (DataStaging::DTR & dtr) [virtual]

Callback called when DTR is finished. This DTR is marked done in the DTR list and if all DTRs for the job have completed, the job is marked as done.

Parameters

| | |
|------------|---|
| <i>dtr</i> | DTR object sent back from the Scheduler |
|------------|---|

5.23.3.5 void DTRGenerator::receiveJob (const JobDescription & job)

A-REX sends data transfer requests to the data staging system through this method. It reads the job.id.input/output files, forms DTRs and sends them to the Scheduler.

Parameters

| | |
|------------|-------------------------|
| <i>job</i> | Job description object. |
|------------|-------------------------|

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

- dtr_generator.h

5.24 DTRInfo Class Reference

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

Public Member Functions

- [DTRInfo](#) (const [JobUsers](#) &users)

5.24.1 Detailed Description

[DTRInfo](#) passes state information from data staging to A-REX via the defined callback, called when the DTR passes to the certain processes. It could for example write to files in the control directory, and this information can be picked up and published by the info system.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 DTRInfo::DTRInfo (const JobUsers & users)

[JobUsers](#) is needed to find the correct control dir

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

- dtr_generator.h

5.25 Entry Class Reference

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

- Entry.h

5.26 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, off_t csize)
- `bool Complete` (void)
- `void Print` (void)
- `void Release` (void)
- `void Remove` (void)

5.26.1 Detailed Description

Representation of delivered file chunks.

5.26.2 Member Function Documentation

5.26.2.1 void ARex::FileChunks::Release (void)

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

5.26.2.2 void ARex::FileChunks::Remove (void)

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

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

- `FileChunks.h`

5.27 ARex::FileChunksList Class Reference

Container for `FileChunks` instances.

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

Public Member Functions

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

5.27.1 Detailed Description

Container for [FileChunks](#) instances.

5.27.2 Member Function Documentation

5.27.2.1 [FileChunks](#)& [ARex::FileChunksList::Get](#) (std::string *path*)

Returns previously created [FileChunks](#) 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.

5.27.2.2 [FileChunks](#)* [ARex::FileChunksList::GetStuck](#) (void)

Returns pointer to first stuck file. File is considered 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](#)

5.28 FileData Class Reference

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

- [info_types.h](#)

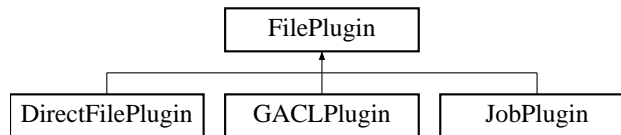
5.29 FileNode Class Reference

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

- [fileroot.h](#)

5.30 FilePlugin Class Reference

Inheritance diagram for FilePlugin:



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

- fileroot.h

5.31 FileRoot Class Reference

Data Structures

- class [ServerParams](#)

Friends

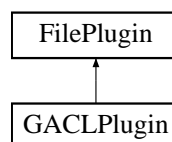
- class [GridFTP_Commands](#)

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

- fileroot.h

5.32 GACLPlugin Class Reference

Inheritance diagram for GACLPlugin:



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

- gacplugin.h

5.33 gm_dirs_ Struct Reference

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

- jobplugin.h

5.34 GEnvironment Class Reference

Public Member Functions

- std::string [nordugrid_loc](#) (void) const
- std::string [nordugrid_data_loc](#) (void) const
- std::string [nordugrid_libexec_loc](#) (void) const
- std::string [nordugrid_config_loc](#) (void) const
- std::string [support_mail_address](#) (void) const

5.34.1 Member Function Documentation

5.34.1.1 std::string GEnvironment::nordugrid_config_loc (void) const

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

5.34.1.2 std::string GEnvironment::support_mail_address (void) const

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

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

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

5.35 gridftpd::GEnvironment 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

5.35.1 Member Function Documentation

5.35.1.1 `std::string gridftp::GMEEnvironment::nordugrid_config_loc (void) const`

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

5.35.1.2 `std::string gridftp::GMEEnvironment::support_mail_address (void) const`

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

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

- `gridftp/conf/environment.h`

5.36 GridFTP_Commands Class Reference

Data Structures

- class `close_semaphore_t`
- struct `data_buffer_t`

Friends

- class `GridFTP_Commands_timeout`

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

- `commands.h`

5.37 GridFTP_Commands_timeout Class Reference

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

- `commands.h`

5.38 ARex::GridManager Class Reference

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

- `grid_manager.h`

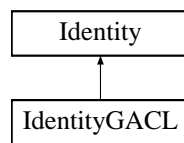
5.39 Hopi::Hopi Class Reference

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

- `hopi.h`

5.40 Identity Class Reference

Inheritance diagram for Identity:



Data Structures

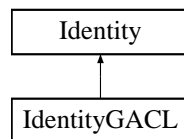
- class [Item](#)

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

- `identity.h`

5.41 IdentityGACL Class Reference

Inheritance diagram for IdentityGACL:

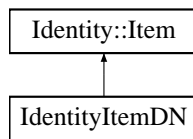


The documentation for this class was generated from the following file:

- `identity_gacl.h`

5.42 IdentityItemDN Class Reference

Inheritance diagram for IdentityItemDN:

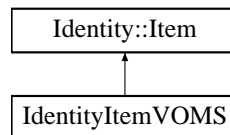


The documentation for this class was generated from the following file:

- identity_dn.h

5.43 IdentityItemVOMS Class Reference

Inheritance diagram for IdentityItemVOMS:



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

- identity_voms.h

5.44 Index Class Reference

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

- Index.h

5.45 ISIS::ISIService Class Reference

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

- isis.h

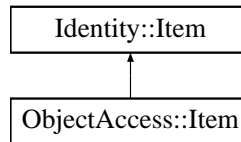
5.46 ISIS::ISISecAttr Class Reference

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

- security.h

5.47 ObjectAccess::Item Class Reference

Inheritance diagram for ObjectAccess::Item:

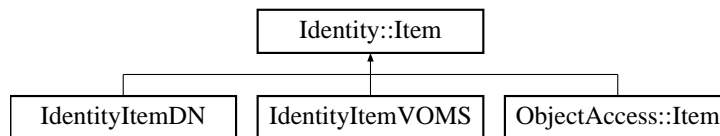


The documentation for this class was generated from the following file:

- `object_access.h`

5.48 Identity::Item Class Reference

Inheritance diagram for Identity::Item:



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

- `identity.h`

5.49 Janitor Class Reference

Class to communicate with [Janitor](#) - Dynmaic Runtime Environment handler.

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

Public Member Functions

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

5.49.1 Detailed Description

Class to communicate with [Janitor](#) - Dynmaic Runtime Environment handler.

5.49.2 Constructor & Destructor Documentation

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

Creates instance representing job entry in [Janitor](#) database.

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

5.49.3 Member Function Documentation

5.49.3.1 bool Janitor::deploy (void)

Registers associated job with [Janitor](#) and deploys dynamic RTEs.

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

5.49.3.2 bool Janitor::remove (void)

Removes job from those handled by [Janitor](#) and releases associated RTEs.

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

5.49.3.3 Result Janitor::result (void)

Returns true if operation initiated by [deploy\(\)](#) or [remove\(\)](#) succeeded.

It should be called after [wait\(\)](#) returned true.

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

Wait till operation initiated by [deploy\(\)](#) or [remove\(\)](#) finished.

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

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

- `janitor.h`

5.50 `job_state_rec_t` Struct Reference

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

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

5.51 `JobDescription` Class Reference

Friends

- class [JobsList](#)

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

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

5.52 `JobLocalDescription` Class Reference

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

- `info_types.h`

5.53 `JobLog` Class Reference

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

5.53.1 Detailed Description

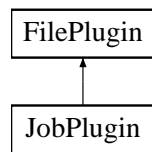
Put short information into log when every job starts/finishes. And store more detailed information for Reporter.

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

- `job_log.h`

5.54 `JobPlugin` Class Reference

Inheritance diagram for `JobPlugin`:



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

- jobplugin.h

5.55 ARex::JobRecord Class Reference

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

- JobRecord.h

5.56 JobsList Class Reference

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

- states.h

5.57 JobsListConfig Class Reference

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

Friends

- class [JobsList](#)

5.57.1 Detailed Description

Class to represent information read from configuration.

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

- states.h

5.58 JobUser Class Reference

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

- [users.h](#)

5.59 JobUserHelper Class Reference

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

- [users.h](#)

5.60 JobUsers Class Reference

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

- [users.h](#)

5.61 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](#) ()

5.61.1 Detailed Description

[LdapQuery](#) class; querying of LDAP servers.

5.61.2 Member Enumeration Documentation

5.61.2.1 enum gridftpd::LdapQuery::Scope

Scope for a LDAP queries. Use when querying.

5.61.3 Constructor & Destructor Documentation

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

Constructs a new [LdapQuery](#) object and sets connection options. The connection is first established when calling Query.

5.61.3.2 `gridftp::LdapQuery::~~LdapQuery ()`

Destructor. Will disconnect from the ldapserver if still connected.

5.61.4 Member Function Documentation

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

Returns the hostname of the ldap-server.

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

Queries the ldap server.

5.61.4.3 `void gridftp::LdapQuery::Result (ldap_callback callback, void * ref) throw (LdapQueryError)`

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

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

- `ldapquery.h`

5.62 gridftp::LdapQueryError Class Reference

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

Public Member Functions

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

5.62.1 Detailed Description

[LdapQuery](#) exception. Gets thrown when an error occurs in a query.

5.62.2 Constructor & Destructor Documentation

5.62.2.1 `gridftp::LdapQueryError::LdapQueryError (std::string message) [inline]`

Standard exception class constructor.

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

- `ldapquery.h`

5.63 `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`

5.64 `gridftp::RunPlugin::lib_plugin_t` Union Reference

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

- `gridftp/run/run_plugin.h`

5.65 `ARex::LoggerClient` Class Reference

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

- `client.h`

5.66 `LRMSResult` Class Reference

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

- `info_types.h`

5.67 `ISIS::Neighbor_Container` Class Reference

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

- [isis.h](#)

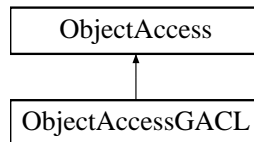
5.68 numvalue_for_shell Class Reference

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

- [job_desc.h](#)

5.69 ObjectAccess Class Reference

Inheritance diagram for ObjectAccess:



Data Structures

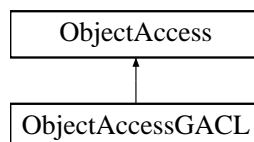
- class [Item](#)

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

- [object_access.h](#)

5.70 ObjectAccessGACL Class Reference

Inheritance diagram for ObjectAccessGACL:



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

- [object_access_gacl.h](#)

5.71 ARex::OptimizedInformationContainer Class Reference

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

- arex.h

5.72 gridftpd::ParallelLdapQueries Class Reference

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

5.72.1 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

5.73 ARex::PayloadBigFile Class Reference

Public Member Functions

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

5.73.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

5.73.1.2 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

5.74 Hopi::PayloadBigFile Class Reference

Public Member Functions

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

5.74.1 Constructor & Destructor Documentation

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

Creates object associated with file for reading from it

5.74.1.2 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

5.75 ARex::PayloadFAFile Class Reference

Public Member Functions

- [PayloadFAFile](#) (Arc::FileAccess *h, Size_t start, Size_t end)

5.75.1 Constructor & Destructor Documentation

5.75.1.1 ARex::PayloadFAFile::PayloadFAFile (Arc::FileAccess * h, Size_t start, Size_t end)

Creates object associated with file for reading from it

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

- a-rex/PayloadFile.h

5.76 Hopi::PayloadFile Class Reference

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

Public Member Functions

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

5.76.1 Detailed Description

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

5.76.2 Constructor & Destructor Documentation

5.76.2.1 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.

5.76.2.2 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

5.77 ARex::PayloadFile Class Reference

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

Public Member Functions

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

5.77.1 Detailed Description

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

5.77.2 Constructor & Destructor Documentation

5.77.2.1 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.

5.77.2.2 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

5.78 DREService::PerlProcessor Class Reference

Data Structures

- struct **ThreadInterface**

Public Member Functions

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

5.78.1 Constructor & Destructor Documentation

5.78.1.1 DREService::PerlProcessor::PerlProcessor (int *threadNumber*, [TaskQueue](#) * *pTaskQueue*, [TaskSet](#) * *pTaskSet*)

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

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

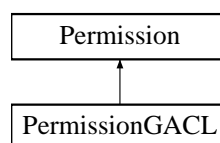
Destructor.

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

- PerlProcessor.h

5.79 Permission Class Reference

Inheritance diagram for Permission:

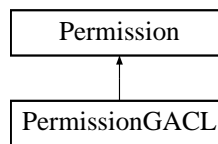


The documentation for this class was generated from the following file:

- permission.h

5.80 PermissionGACL Class Reference

Inheritance diagram for PermissionGACL:



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

- permission_gacl.h

5.81 Policy Class Reference

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

- Policy.h

5.82 ArcSec::Charon::PolicyLocation Class Reference

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

- charon.h

5.83 ContinuationPlugins::result_t Class Reference

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

- plugins.h

5.84 RunFunction Class Reference

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

- run_function.h

5.85 RunParallel Class Reference

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

- `run_parallel.h`

5.86 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`

5.87 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`

5.88 RunPlugins Class Reference

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

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

5.89 RunRedirected Class Reference

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

- `run_redirected.h`

5.90 Server Class Reference

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

- Server.h

5.91 FileRoot::ServerParams Class Reference

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

- fileroot.h

5.92 ArcSec::Service_AA Class Reference

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

5.92.1 Detailed Description

A Service which includes the AttributeAuthority functionality; it accepts the <sample: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 <sample:Assertion> which includes the <Attribute>

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

- aaservice.h

5.93 Arc::Service_JavaWrapper Class Reference

Public Member Functions

- virtual Arc::MCC_Status [process](#) (Arc::Message &, Arc::Message &)

5.93.1 Member Function Documentation

5.93.1.1 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

5.94 Arc::Service_PythonWrapper Class Reference

Public Member Functions

- virtual Arc::MCC_Status [process](#) (Arc::Message &, Arc::Message &)

5.94.1 Member Function Documentation

5.94.1.1 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

5.95 ArcSec::Service_SLCS Class Reference

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

5.95.1 Detailed Description

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

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

- slcs.h

5.96 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 &)

5.96.1 Detailed Description

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

5.96.2 Constructor & Destructor Documentation

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

Constructor

5.96.3 Member Function Documentation

5.96.3.1 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

5.97 SimpleMap Class Reference

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

- simplemap.h

5.98 DREService::Task Class Reference

Public Member Functions

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

5.98.1 Constructor & Destructor Documentation

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

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

5.98.1.2 virtual DREService::Task::~~Task (void) [virtual]

Destructor.

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

- Task.h

5.99 DREService::TaskQueue Class Reference

Public Member Functions

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

5.99.1 Constructor & Destructor Documentation

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

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

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

Destructor.

5.99.2 Member Function Documentation

5.99.2.1 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.

5.99.2.2 [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

5.100 DREService::TaskSet Class Reference

Public Member Functions

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

5.100.1 Constructor & Destructor Documentation

5.100.1.1 DREService::TaskSet::TaskSet (int size)

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

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

Destructor.

5.100.2 Member Function Documentation

5.100.2.1 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

5.101 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

5.102 gridftp2::UrlMapConfig Class Reference

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

- gridftp2/conf/conf_map.h

5.103 UrlMapConfig Class Reference

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

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

5.104 userspec_t Class Reference

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

- userspec.h

5.105 value_for_shell Class Reference

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

- job_desc.h

5.106 voms Struct Reference

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

Data Fields

- std::string [server](#)
- std::string [vname](#)
- std::vector< [voms_attrs](#) > [attrs](#)

5.106.1 Detailed Description

VOMS data

5.106.2 Field Documentation

5.106.2.1 `std::vector<voms_attrs> voms::attrs`

User's characteristics

5.106.2.2 `std::string voms::server`

The VOMS server DN, as from its certificate

5.106.2.3 `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`

5.107 `voms_attrs` Struct Reference

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

Data Fields

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

5.107.1 Detailed Description

VOMS attributes

5.107.2 Field Documentation

5.107.2.1 `std::string voms_attrs::cap`

user's capability

5.107.2.2 `std::string voms_attrs::group`

user's group

5.107.2.3 `std::string voms_attrs::role`

user's role

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

- `auth.h`

5.108 ZeroUInt Class Reference

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

5.108.1 Detailed Description

[ZeroUInt](#) 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`

Index

- ~CacheService
 - Cache::CacheService, [19](#)
- ~DREWebService
 - DREService::DREWebService, [25](#)
- ~DTRGenerator
 - DTRGenerator, [26](#)
- ~LdapQuery
 - gridftpd::LdapQuery, [41](#)
- ~PayloadBigFile
 - ARex::PayloadBigFile, [44](#)
 - Hopi::PayloadBigFile, [45](#)
- ~PayloadFile
 - ARex::PayloadFile, [46](#)
 - Hopi::PayloadFile, [46](#)
- ~PerlProcessor
 - DREService::PerlProcessor, [47](#)
- ~Task
 - DREService::Task, [52](#)
- ~TaskQueue
 - DREService::TaskQueue, [53](#)
- ~TaskSet
 - DREService::TaskSet, [54](#)
- Arc::Service_JavaWrapper, [50](#)
 - process, [50](#)
- Arc::Service_PythonWrapper, [51](#)
 - process, [51](#)
- ArcSec::Charon, [21](#)
- ArcSec::Charon::PolicyLocation, [48](#)
- ArcSec::Service_AA, [50](#)
- ArcSec::Service_SLCS, [51](#)
- ARex::ARexGMConfig, [13](#)
- ARex::ARexJob, [13](#)
 - ARexJob, [14](#)
 - Cancel, [14](#)
 - ChooseSessionDir, [14](#)
 - Clean, [14](#)
 - CreateFile, [15](#)
 - Failed, [15](#)
 - Failure, [15](#)
 - GetDescription, [15](#)
 - ID, [15](#)
 - Jobs, [15](#)
 - LogDir, [15](#)
 - LogFiles, [15](#)
 - OpenDir, [15](#)
 - OpenFile, [15](#)
 - OpenLogFile, [16](#)
 - Resume, [16](#)
 - SessionDir, [16](#)
 - State, [16](#)
 - TotalJobs, [16](#)
 - UpdateCredentials, [16](#)
- ARex::ARexService, [16](#)
- ARex::CountedResource, [22](#)
- ARex::FileChunks, [29](#)
 - Release, [29](#)
 - Remove, [29](#)
- ARex::FileChunksList, [29](#)
 - Get, [30](#)
 - GetStuck, [30](#)
- ARex::GridManager, [33](#)
- ARex::JobRecord, [39](#)
- ARex::LoggerClient, [42](#)
- ARex::OptimizedInformationContainer, [44](#)
- ARex::PayloadBigFile, [44](#)
 - ~PayloadBigFile, [44](#)
 - PayloadBigFile, [44](#)
- ARex::PayloadFAFile, [45](#)
 - PayloadFAFile, [45](#)
- ARex::PayloadFile, [46](#)
 - ~PayloadFile, [46](#)
 - PayloadFile, [46](#)
- ARexJob
 - ARex::ARexJob, [14](#)
- attrs
 - voms, [56](#)
- AuthEvaluator, [17](#)
- AuthUser, [17](#)
- AuthVO, [17](#)
- Cache::CacheService, [19](#)

- ~CacheService, [19](#)
- CacheCheck, [20](#)
- CacheLink, [20](#)
- CacheService, [19](#)
- operator bool, [20](#)
- process, [20](#)
- RegistrationCollector, [20](#)
- CacheCheck
 - Cache::CacheService, [20](#)
- CacheConfig, [17](#)
 - CacheConfig, [18](#)
 - parseINICnf, [18](#)
 - setCacheDirs, [18](#)
- CacheConfigException, [18](#)
- CacheLink
 - Cache::CacheService, [20](#)
- CacheService
 - Cache::CacheService, [19](#)
- Cancel
 - ARex::ARexJob, [14](#)
- cancelJob
 - DTRGenerator, [27](#)
- cap
 - voms_attrs, [56](#)
- checkUploadedFiles
 - DTRGenerator, [27](#)
- ChooseSessionDir
 - ARex::ARexJob, [14](#)
- Clean
 - ARex::ARexJob, [14](#)
- CommFIFO, [21](#)
- ConfigSections, [21](#)
- ContinuationPlugins, [22](#)
- ContinuationPlugins::result_t, [48](#)
- CreateFile
 - ARex::ARexJob, [15](#)
- DataStaging::DataDeliveryService, [22](#)
- deploy
 - Janitor, [37](#)
- DirectAccess, [23](#)
- DirectAccess::diraccess_t, [23](#)
- DirectFilePlugin, [24](#)
- DirEntry, [24](#)
- DREService, [11](#)
- DREService::DREWebService, [24](#)
 - ~DREWebService, [25](#)
 - DREWebService, [25](#)
 - logger, [25](#)
 - makeFault, [25](#)
 - ns_, [25](#)
 - process, [25](#)
- DREService::PerlProcessor, [47](#)
 - ~PerlProcessor, [47](#)
 - PerlProcessor, [47](#)
- DREService::Task, [52](#)
 - ~Task, [52](#)
 - Task, [52](#)
- DREService::TaskQueue, [53](#)
 - ~TaskQueue, [53](#)
 - pushTask, [53](#)
 - shiftTask, [53](#)
 - TaskQueue, [53](#)
- DREService::TaskSet, [54](#)
 - ~TaskSet, [54](#)
 - removeTask, [54](#)
 - TaskSet, [54](#)
- DREWebService
 - DREService::DREWebService, [25](#)
- DTRGenerator, [26](#)
 - ~DTRGenerator, [26](#)
 - cancelJob, [27](#)
 - checkUploadedFiles, [27](#)
 - DTRGenerator, [26](#)
 - queryJobFinished, [27](#)
 - receiveDTR, [27](#)
 - receiveJob, [27](#)
- DTRInfo, [28](#)
 - DTRInfo, [28](#)
- Entry, [28](#)
- Failed
 - ARex::ARexJob, [15](#)
- Failure
 - ARex::ARexJob, [15](#)
- FileData, [30](#)
- FileNode, [30](#)
- FilePlugin, [31](#)
- FileRoot, [31](#)
- FileRoot::ServerParams, [50](#)
- GACLPlugin, [31](#)
- Get
 - ARex::FileChunksList, [30](#)
- GetDescription
 - ARex::ARexJob, [15](#)
- GetStuck
 - ARex::FileChunksList, [30](#)
- gm_dirs_, [32](#)

- GMEEnvironment, 32
 - nordugrid_config_loc, 32
 - support_mail_address, 32
- GridFTP_Commands, 33
- GridFTP_Commands_timeout, 33
- gridftpd::ConfigSections, 21
- gridftpd::Daemon, 22
- gridftpd::GMEEnvironment, 32
 - nordugrid_config_loc, 33
 - support_mail_address, 33
- gridftpd::LdapQuery, 40
 - ~LdapQuery, 41
 - Host, 41
 - LdapQuery, 41
 - Query, 41
 - Result, 41
 - Scope, 40
- gridftpd::LdapQueryError, 41
 - LdapQueryError, 42
- gridftpd::ParallelLdapQueries, 44
- gridftpd::RunPlugin, 49
- gridftpd::RunPlugin::lib_plugin_t, 42
- gridftpd::UrlMapConfig, 55
- group
 - voms_attrs, 56
- Hopi::Hopi, 34
- Hopi::PayloadBigFile, 45
 - ~PayloadBigFile, 45
 - PayloadBigFile, 45
- Hopi::PayloadFile, 45
 - ~PayloadFile, 46
 - PayloadFile, 46
- Host
 - gridftpd::LdapQuery, 41
- ID
 - ARex::ARexJob, 15
- Identity, 34
- Identity::Item, 36
- IdentityGACL, 34
- IdentityItemDN, 34
- IdentityItemVOMS, 35
- Index, 35
- ISIS::ISISService, 35
- ISIS::ISISecAttr, 35
- ISIS::Neighbor_Container, 42
- Janitor, 36
 - deploy, 37
 - Janitor, 37
 - remove, 37
 - result, 37
 - wait, 37
- job_state_rec_t, 38
- JobDescription, 38
- JobLocalDescription, 38
- JobLog, 38
- JobPlugin, 38
- Jobs
 - ARex::ARexJob, 15
- JobsList, 39
- JobsListConfig, 39
- JobUser, 39
- JobUserHelper, 40
- JobUsers, 40
- LdapQuery
 - gridftpd::LdapQuery, 41
- LdapQueryError
 - gridftpd::LdapQueryError, 42
- LogDir
 - ARex::ARexJob, 15
- LogFiles
 - ARex::ARexJob, 15
- logger
 - DREService::DREWebService, 25
- LRMSResult, 42
- makeFault
 - DREService::DREWebService, 25
- nordugrid_config_loc
 - GMEEnvironment, 32
 - gridftpd::GMEEnvironment, 33
- ns_
 - DREService::DREWebService, 25
- numvalue_for_shell, 43
- ObjectAccess, 43
- ObjectAccess::Item, 36
- ObjectAccessGACL, 43
- OpenDir
 - ARex::ARexJob, 15
- OpenFile
 - ARex::ARexJob, 15
- OpenLogFile
 - ARex::ARexJob, 16
- operator bool
 - Cache::CacheService, 20

- parseINIConf
 - CacheConfig, 18
- PayloadBigFile
 - ARex::PayloadBigFile, 44
 - Hopi::PayloadBigFile, 45
- PayloadFAFile
 - ARex::PayloadFAFile, 45
- PayloadFile
 - ARex::PayloadFile, 46
 - Hopi::PayloadFile, 46
- PerlProcessor
 - DREService::PerlProcessor, 47
- Permission, 47
- PermissionGACL, 48
- Policy, 48
- process
 - Arc::Service_JavaWrapper, 50
 - Arc::Service_PythonWrapper, 51
 - Cache::CacheService, 20
 - DREService::DREWebService, 25
 - SPService::Service_SP, 52
- pushTask
 - DREService::TaskQueue, 53
- Query
 - gridftp::LdapQuery, 41
- queryJobFinished
 - DTRGenerator, 27
- receivedTR
 - DTRGenerator, 27
- receiveJob
 - DTRGenerator, 27
- RegistrationCollector
 - Cache::CacheService, 20
- Release
 - ARex::FileChunks, 29
- Remove
 - ARex::FileChunks, 29
- remove
 - Janitor, 37
- removeTask
 - DREService::TaskSet, 54
- Result
 - gridftp::LdapQuery, 41
- result
 - Janitor, 37
- Resume
 - ARex::ARexJob, 16
- role
 - voms_attrs, 56
- RunFunction, 48
- RunParallel, 49
- RunPlugin, 49
- RunPlugin::lib_plugin_t, 42
- RunPlugins, 49
- RunRedirected, 49
- Scope
 - gridftp::LdapQuery, 40
- Server, 50
- server
 - voms, 56
- Service_SP
 - SPService::Service_SP, 52
- SessionDir
 - ARex::ARexJob, 16
- setCacheDirs
 - CacheConfig, 18
- shiftTask
 - DREService::TaskQueue, 53
- SimpleMap, 52
- SPService::Service_SP, 51
 - process, 52
 - Service_SP, 52
- State
 - ARex::ARexJob, 16
- support_mail_address
 - GMEnvironment, 32
 - gridftp::GMEnvironment, 33
- Task
 - DREService::Task, 52
- TaskQueue
 - DREService::TaskQueue, 53
- TaskSet
 - DREService::TaskSet, 54
- TotalJobs
 - ARex::ARexJob, 16
- UnixMap, 54
- UpdateCredentials
 - ARex::ARexJob, 16
- UrlMapConfig, 55
- userspec_t, 55
- value_for_shell, 55
- voms, 55
 - attrs, 56
 - server, 56

- voname, [56](#)
- voms_attrs, [56](#)
 - cap, [56](#)
 - group, [56](#)
 - role, [56](#)
- voname
 - voms, [56](#)
- wait
 - Janitor, [37](#)
- ZeroUInt, [57](#)