



DataGrid

Installation Guide

Integration Team (WP6)

Document identifier:	DataGrid-06-TED-0105-2-0
Date:	October 29, 2003
Workpackage:	Integration Team (WP6)
Partners:	Contributions from all partners

Abstract: This document describes the installation procedures for the EDG 2.0.x software. Site administrators providing resources to the EDG testbeds are the primary audience; however, others may find the detailed information presented here useful. This document assumes that the site administrator will use LCFGng to install the software. It is hoped that a future version of this guide will also include instructions for the manual installation of the software.



Change Log

Version	Date	Comment
1	31 Dec 2003	First public version
2	12 Jan 2003	Updated version.
3	24 Oct 2003	Updated version to EDG-2.0.
3	29 Oct 2003	First EDG 2.0.x draft



Contents

1 Overview	6
1.1 Information System	6
1.2 Workload Management	8
1.3 Data Management	8
1.4 Security	9
1.5 Resources	10
1.6 Monitoring	10
1.7 Miscellaneous Services	11
2 Release Management	12
2.1 Obtaining EDG Releases	12
2.2 EDG Versioning	12
2.3 Release Contents	13
3 Installation	15
3.1 Manual Installation	15
3.2 LCFGng Installation	15
3.3 UML-LCFGng Installation	17
4 Authentication, Authorization, and Security	18
4.1 Access Policy	18
4.2 Authentication	18
4.3 Authorization	20
4.4 Firewalls	22
4.5 Miscellaneous Comments	22
5 EDG Machine Configuration	24
5.1 Manual Installation of User Interface	24
6 EDG Service Configuration	26
6.1 Berkeley Database Information Index (BDII)	26
6.2 EDG Gatekeeper	28
6.3 GOUT	30
6.4 TCP Throughput Measurement (IPerf)	31
6.5 Job Controller (edg-wl-jc)	32
6.6 Logging & Bookkeeping Server (edg-wl-lb)	33
6.7 Fabric Monitoring (LEMON)	34



6.8	Local Logger	35
6.9	Local Replica Catalog (LRC)	36
6.10	Network Server (edg-wl-ns)(MON??)	38
6.11	Probes Coordination Protocol Daemon (PCPD)	40
6.12	Relational Grid Monitoring Architecture (R-GMA)	41
6.13	Replica Manager (RM)	43
6.14	Replica Location Index (RLI)	45
6.15	Replica Metadata Catalog (RMC)	46
6.16	Replica Optimization Service (ROS)	48
6.17	Runtime Control System (RTCS)	50
6.18	Storage Element Web Service (SEWS)	51
6.19	UDPMon	53
6.20	Virtual Organization Membership Service (VOMS)	54
6.21	Workload Manager (edg-wl-wm)(RB??)	58
7	Globus Service Configuration	60
7.1	GridFTP	60
7.2	Monitoring and Discovery Service (MDS)	62
8	Third-party Service Configuration	63
8.1	CondorG	63
8.2	Mercury	64
8.3	MyProxy	65
8.4	MySQL	66
8.5	Remote File IO (RFIO)	68
8.6	Tomcat Daemon	69
9	Testing Your Installation	70
10	Support	71
10.1	Contacts	71
10.2	Website	71
11	Appendix	72
11.1	Time Synchronization	72
11.2	Monitoring	73
11.3	Installing Application Software	73
11.4	GSI FTP Daemon	73
12	Portable Batch System	75
12.1	Configuration	75
12.2	Testing	77



List of Tables

2.1	Standard Machine Types	13
4.1	Daemon Ports and Machine Types	23
6.1	BDII Installation Information	26
6.2	EDG Gatekeeper Installation Information	29
6.3	GOUT Installation Information	30
6.4	IPerf Installation Information	31
6.5	Job Controller Installation Information	32
6.6	L&B Server Installation Information	33
6.7	Fabric Monitoring (LEMON) Installation Information	34
6.8	Local Logger Installation Information	35
6.9	LRC Installation Information	36
6.10	Network Server Installation Information	38
6.11	PCPD Installation Information	40
6.12	R-GMA Installation Information	41
6.13	RMC Installation Information	46
6.14	ROS Installation Information	48
6.15	RTCS Installation Information	50
6.16	Storage Element Web Service Installation Information	51
6.17	UDPMon Installation Information	53
6.18	VOMS Installation Information	54
6.19	Workload Manager Installation Information	58
7.1	GridFTP Installation Information	60
7.2	MDS Installation Information	62
8.1	CondorG Installation Information	63
8.2	Mercury Installation Information	64
8.3	MyProxy Installation Information	65
8.4	MySQL Installation Information	66
8.5	RFIO Installation Information	68
8.6	Tomcat Installation Information	69
11.1	Spec File Changes	73
11.2	FTP Daemon Parameters	74
12.1	PBS Installation Information	75

1 Overview

The European DataGrid (EDG) operates an application testbed which allows users from various application areas to test the EDG software in a semi-production environment. Volunteer sites provide the hardware which run the necessary EDG services. This manual describes the installation of the EDG software to allow your site to join the application testbed.

In parallel, EDG operates a development testbed to test new releases of the EDG software before deployment on the application testbed. The descriptions in this manual are likely to lag behind the software in the development release, but is nonetheless useful.

The following overview of the software is from the point-of-view of one trying to deploy the services. For a more functional overview, see the EDG Users' Guide.

Cautions There is a large amount of information in this manual and important points for a successful installation are spread throughout its entirety. A few facts about the software are collected here to help you determine if your site is a suitable candidate for running the EDG software.

- The EDG software is tested on RedHat Linux 7.3. The software is likely to run with simple modifications on more recent versions of RedHat; however, EDG has not verified the operation of the software on those versions. The software may work on non-linux operating systems, but will likely involve significant effort for the person porting the software.
- These installation instructions assume that you use LCFGng for installation. While manual installation is possible, to date no one has successfully configured a full site manually.
- An absolutely minimal site consists of four machines. An LCFGng server, gatekeeper (CE), storage element (SE), and monitoring machine (MON) which runs the site R-GMA services. (Although sites providing only computational or storage services are possible.)
- To meaningfully contribute to the EDG testbed, your machines (services) must have cryptographic certificates signed by an EDG-recognized certificate authority.
- Site administrators with services in the EDG testbeds are expected to provide first-level support to your local users and to be responsive to any reported problems with your deployed services.

Even if your site meets the above requirements, you are strongly suggested to read through the manual before starting to avoid unpleasant surprises during the installation.

1.1 Information System

The information system forms the backbone of the grid. It provides information about service addresses, capabilities, and states. It also provides a means of querying the information to allow service discovery and service requirement matching.

There are two separate information systems which can be used to tie together the other EDG services—MDS from Globus and R-GMA from EDG. Both use a common information schema and a common set of “information provider” scripts on the grid resources to provide the information.

The architectures of the two systems are quite different and expose different query languages to the information system user. Consequently the two systems may be run in parallel, but the two systems are not interoperable. The EDG testbeds primarily use R-GMA, but basic MDS functionality is also deployed in parallel.

1.1.1 Relational Grid Monitoring Architecture (R-GMA)

The Relational Grid Monitoring Architecture (R-GMA) is, as the name implies, an implementation of the Grid Monitoring Architecture proposed by the Global Grid Forum based on a relational database model. As R-GMA is based on a relational model, it is not surprising that the query language exposed to the users is SQL-like.

At the highest level is an Information Catalog (IC) which acts as a registry for producers of information, i.e. grid resources. It is truly a registry containing only schema for available information and the location of the producers of that information. Currently there is a single IC per grid. Redundant replicated catalogs are expected in a future release of the software.

At the site level, an R-GMA server caches the resource information generated at the site and registers it's availability to the IC. This acts as the site's gateway to the information system while at the same time being the contact point for outside queries. One such service is required per site. (It is deployed by default on the "MON" machine type.)

At the resource level, R-GMA uses GIN (Gadget IN) to run periodically the standard information provider scripts and to publish this information to the site's R-GMA server.

The resource broker described later requires a fast, robust cache of the information system's data accessible via LDAP queries. GOUT (Gadget OUT) provides this cache for R-GMA. Usually the GOUT server is started only for sites running a resource broker.

Deployment : Single IC per testbed. One R-GMA server per site. One GOUT service on sites deploying a resource broker.

Function : Information system based on a relational database model. Used for resource discovery and basic resource requirements matching.

State : Deployed as primary information system. Working with some stability and scalability issues.

1.1.2 Monitoring and Discovery Service (MDS)

The Monitoring and Discovery Service (MDS) from Globus fills the same niche as R-GMA but is instead based on LDAP. It is the older of the two and was used exclusively in previous releases of the EDG software.

On the resources, MDS calls the standard information provider scripts and publishes the obtained information into a local LDAP server (called a GRIS-Grid Resource Information Service). Each site runs another LDAP server (GIIS-Grid Index Information Service) to which all of the site's GRISes register. The site GIIS uses cached information to respond to queries when possible, contacting the GRISes only to refresh stale information.

Above the sites, there is an arbitrarily large hierarchy of GIISes to collect information from sites in the grid. This hierarchy is capped by a single GIIS which collects information from the grid as a whole. This is the access point for queries of the information system.

The top-level GIIS unfortunately could not handle the load placed on it by the resource broker(s). Therefore, EDG configured a standard OpenLDAP server to act as the fast, robust cache for the resource broker. This was dubbed the BDII-Berkeley Database Information Index. One of these is deployed per resource broker and caches the information available from the top-level GIIS.

Currently the MDS system is deployed in parallel with R-GMA, but it is not actively used. The only part of MDS required by the current EDG software is the GRIS running on the computing elements.

Deployment : One GRIS on each resource; one GIIS per site. One top-level GIIS per grid. One BDII per deployed resource broker.

Function : Information system based on a LDAP database model. Used for resource discovery and basic resource requirements matching.

State : Deployed as secondary information system; not actively used. GRIS on computing element is only required element for current EDG release. Working.

1.2 Workload Management

The workload management system is a collection of services designed to give the end user high-level access to the computational resources of the grid. The workload management system, from a user's perspective, is similar to that of a standard batch system, providing commands to submit a job, check its status, manage its state, and retrieve its output.

The workload management client commands are deployed on the "User Interface" (UI) machine and are designed to be portable and lightweight. The consistent state of the users jobs can be accessed from any machine with the workload management user interface software installed. The UI machine also contains the client software for all of the other EDG services.

The Resource Broker (RB) machine is the workhorse of the workload management system, containing the daemons necessary to interact with the user interface, perform matching between job requirements and available resources, and steer the jobs through their complete lifecycle on remote computing resources. The input and output of the managed jobs are also cached on the RB machine. To make use of the workload management services at least one must be deployed per grid, but typically several are deployed to reduce the load on any one broker.

Running a resource broker implies also running a GOUT service (or BDII) on the same site and optionally a MyProxy server.

Deployment : User Interface, typically one to several per site. Resource Broker, typically several per grid.

Function : UI collects all EDG client software and acts as users' gateway to the grid. RB manages the lifecycle of user jobs from submission through to completion.

State : Deployed; working.

1.3 Data Management

The EDG data management software concerns the management of the data files and associated metadata for a virtual organization. One aspect involves the deployment of a distributed database which contains the locations of the data files linked with another database contains the associated metadata. The Replica Manager (RM) tools contact these databases and interact with individual resources to allow the user high-level data replication services. The RM tools are deployed where users will need them (user interface machines and worker nodes).

The EDG implementation of the Replica Location Service (RLS) consists of two webservises running as servlets in a tomcat container.¹ These two services are the Local Replica Catalog (LRC) and the Replica Location Index (RLI). As currently deployed, the LRC is run as a centralized service (one per virtual organization) and RLI is not deployed. The RLI will allow this to be a truly distributed service in the future.

Two associated servlets are the Replica Optimization Service (ROS) and the Replica Metadata Catalog (RMC). ROS helps the replica manager to pick the best replica for a particular operation; there is one

¹There is a Globus installation of RLS as well. The EDG and Globus versions of RLS are not currently interoperable.

server deployed per site (on the MON node). The RMC is a centralized service for a virtual organization; it is deployed on the same node as the LRC for that virtual organization.

Deployment : RM on user interface machines and worker nodes. LRC and RMC on centralized server; one per VO. ROS on MON node; one per site.

Function : Provides database of available files and replicas for a virtual organization. RM tools provide high-level interface to databases and underlying storage resources.

State : Deployed with central LRC; working. RLI software distributed as part of release but not configured.

1.4 Security

Allowing legitimate users access to the grid's services and resources while barring unauthorized access is of paramount importance. The EDG security model uses the Globus Security Infrastructure (GSI) for the underlying security technology. GSI is an implementation of a Public Key Infrastructure (PKI).

Users and hosts (services) are authenticated based on cryptographic certificates issued by Certificate Authorities (CA). There are now more than twenty EDG-approved CAs whose service areas cover Europe, the United States, and some of Asia. These professionally-run services are a vital part of the EDG security infrastructure. Running a CA is a significant investment; fortunately most users and sites are covered by an existing CA and will not have to start one.

A separate infrastructure handles the authorization of users. Currently authorization is handled crudely based on membership in a recognized virtual organization. To support this, a set of "membership servers" list the subject names of members of each VO.

An older server based on LDAP is still in use on the EDG testbeds. A newer version with much enhanced functionality (and allowing eventually for more fine-grained access control) is being phased-in. Currently the only VOs with a newer-VOMS (Virtual Organization Membership Service) server are the WP6 and ITeam VOs. Typical sites will not have to run any of these servers.

Deployment : VO LDAP server; one per virtual organization. Alternatively, one VOMS server per VO.

Function : Provide list of valid users for a particular VO. VOMS provides authorization information to users through non-critical extensions of the users' proxies.

State : Both types deployed and working.

Users (and users' jobs) actually use time-limited proxies generated from the public and private keys of the users. Some actions on the grid (e.g. long-lived jobs) may continue longer than the validity of the proxy. To allow the proxy to be renewed, a MyProxy server will refresh a proxy. The proxy renewal is handled by the resource broker and hence a MyProxy server is usually deployed along with a RB.

Deployment : Somewhere between one per testbed and one per RB. The server must be configured to recognize a given set of RBs.

Function : Allow renewal of user proxies for long-lived jobs.

State : Deployed; proxy renewal is working.

1.5 Resources

Most every site has computational and storage resources to make available to grid users. The computational resources are made available to the grid users by running a gatekeeper (GK) as an interface to a conventional batch system (PBS, LSF, etc). Similarly, the Storage Element (SE) provides consistent grid interface to disk or tape storage.

1.5.1 Computing Element (CE)

A Computing element consists of a gatekeeper and an associated batch system with a number of worker nodes. The gatekeeper is the interface to the grid world. The worker nodes have the client interfaces to the grid services installed, but are in no other way special; they just provide compute cycles for submitted jobs.

Globus provides a standard gatekeeper. EDG has extended the globus gatekeeper to provide hooks to the EDG security framework. The services behind the callouts are the Local Centre Authorization Service (LCAS) and Local Credential Mapping Service (LCMAPS). Both allow more fine-grained access control than permitted by the standard gatekeeper. These can also make use of the new proxies generated by the VOMS servers.

For sites running PBS, there is also an alternate scheduling system based on MAUI called the Run-Time Control System (RTCS). The software is distributed along with the release; however, the standard configuration has this turned off. Turning it on is a simple change in the configuration files.

Deployment : Gatekeeper; one per interfaced batch system. Worker nodes; many per gatekeeper.

Function : Acts as gatekeeper to local computing resources with callouts to support the EDG security framework. For PBS, an alternate scheduling system can be configured as well.

State : Deployed; working.

1.5.2 Storage Element (SE)

Just as for computational resources, there are standard grid interfaces for storage. In previous releases, a Storage Element (SE) was simply a GridFTP server with some interface scripts for Mass Storage Systems (MSS). This type of SE can still be deployed and is now referred to as an "SE CLASSIC".

With EDG release 2.0.x, there is now a storage interface which is implemented as a web service. Access to a SE of this type is much different than for an SE CLASSIC, but for users the differences are masked by the replica manager interface.

The site administrator may choose to deploy either type of storage element; the newer SE with the web service interface is the recommended one.

Deployment : One or more per site.

Function : Interface to (semi-)permanent storage and MSS.

State : Deployed; SE working at basic level. Many bug fixes and improvements will only appear in EDG 2.1.x releases.

1.6 Monitoring

Monitoring is an overly broad term which typically means different things depending on your relationship to the grid. Because different people which to monitor different things, there are several different monitoring packages. The following sections describe them.

1.6.1 Fabric Monitoring (LEMON)

To help a site administrator monitor the state of the local resources (fabric), EDG supplies a fabric monitoring service (LEMON). In the default configuration, monitoring agents run on all nodes send data (performance metrics and daemon status) to the fabric monitoring server (which resides on the MON node). There is a rudimentary web interface to the information. For access to the detailed performance metrics a C API can be used.

Deployment : Monitoring agents on all nodes; one monitoring server per site (on MON node).

Function : Collects performance metrics and provides rudimentary web interface to information.

State : Deployed; working.

1.6.2 Network Monitoring (NM)

The efficient transport of large data files is an important goal for the EDG project; consequently, knowledge of the state of the network is vital. Each site may optionally deploy a Network Monitoring (NM) node. This node is used to gather network statistics between your site and other sites in the grid. This information (along with information from real data transfers) is used as input to a “network cost function” to allow the replica manager to choose the replica of a particular file with the lowest cost (typically the lowest latency).

Deployment : Optionally one per site.

Function : Runs network monitoring software.

State : Deployed; working.

1.6.3 Application Monitoring

The EDG software also contains a specialized package, GRM/PROVE, to collect performance metrics from parallelized programs running in the grid and then to visualize that information. This facility requires that the application be specially-instrumented to use these features.

This software is deployed with the standard release and configured. The services reside on the worker nodes and on the gatekeepers.

1.7 Miscellaneous Services

1.7.1 Spitfire

Spitfire provides a consistent grid interface to relational databases. This interface can easily be added to existing databases. As databases are typically application-specific, these types of nodes are deployed only by virtual organizations.

Deployment : Optionally one per interfaced relational database.

Function : Provides secure grid interface to relational database.

State : Deployed by virtual organizations wanting to “grid-enable” a database.

2 Release Management

As can be seen from the overview, the EDG contains a large number of independent packages which in turn have a large number of dependencies on external and system software. Consequently, a complete EDG release must specify all of the packages and exact versions of those packages used. This is done through the CVS module “edg-release”.

2.1 Obtaining EDG Releases

All of the EDG source code can be obtained from the CVS repository; anonymous access is permitted. More importantly in the present context, the rpm lists and configuration files for all of the EDG releases can also be checked out. A complete guide is available from the web ([html](#)¹, [pdf](#)²). To just checkout the module, put the following into the file `$HOME/.ssh/config`:

```
Host datagrid.in2p3.fr
Protocol 1
Port 2222
PasswordAuthentication yes
RSAAuthentication no
Compression no
User anoncvs
```

then set the `CVS_RSH` variable to point to the `ssh` executable:

```
export CVS_RSH='which ssh'
```

for sh-type shells. Lastly do the checkout itself:

```
cvs -d datagrid.in2p3.fr:/cvs co -r v2_0_0 edg-release
```

This will put the all of the files for release “v2.0.0” into the `edg-release` subdirectory.

2.2 EDG Versioning

The versions of the EDG releases correspond directly to a CVS tag. For example, EDG release 1.4.3 corresponds to tag `v1.4.3` and EDG release 2.0.0 to tag `v2_0_0`. The first digit corresponds to major changes in the features of the EDG code. Changes in the second digit indicate incompatible changes in the EDG software. The last digit marks compatible patches to previous releases.

To determine which version to install, contact the integration team³. This manual describes the installation of the EDG 2.0.x series of releases.

You can find the full list of tags by doing the following:

```
cvs status -v README
```

¹<http://datagrid.in2p3.fr/HOWTO/CCIN2P3CVSaccessHOWTO.html>

²<http://datagrid.in2p3.fr/HOWTO/CCIN2P3CVSaccessHOWTO.pdf>

³<mailto:hep-proj-grid-integration-team@cern.ch>

Table 2.1: Standard Machine Types

RPM List	Configuration	Description
BASE-rpm		Base RH7.3 system and updates
BDII-rpm	BDII-cfg.h	Berkeley Database Information Index
CE-rpm	ComputingElement-cfg.h	Computing Element
IC-rpm	InformationCatalogue-cfg.h	Information Catalog (R-GMA)
lcfgng-server-rpm		Dummy list for LCFGng server
MDS-rpm	MDS-cfg.h	MDS Top-level GIIS
MON-rpm	Monitoring-cfg.h	Site R-GMA server & Fab. monitoring
NM-rpm	NetworkMonitor-cfg.h	Network monitoring
PX-rpm	Proxy-cfg.h	MyProxy server
RB-rpm	ResourceBroker-cfg.h	Resource broker
RLS-rpm	ReplicaLocationService-cfg.h	VO's LRC & RMC
SECLASSIC-rpm	StorageElement-cfg.h	GridFTP-only storage element
SE-rpm	StorageElement-cfg.h	Webservice-accessible storage element
UI-rpm	UserInterface-cfg.h	User interface for access to grid
VOMS-rpm	VOMS-cfg.h	Virtual Organization Membership Server
WN-rpm	WorkerNode-cfg.h	Batch system client

in the `edg-release` directory. All of the tags beginning with “v” are production releases (or varying quality). Those beginning with “b” are branches of the releases. Those beginning with “t” are test tags used in the integration and validation process. Only tags with a “v” prefix should be deployed on the application testbed.

Alternatively you can point your web browser to <http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/edg-release> and check the available tags in the select box on the bottom of the page

2.3 Release Contents

The check out of the release will create three subdirectories under `edg-release`—`ng_rpmlist`, `ng_source`, and `ng_updaterep`. These contain the packages (and versions) used in a given EDG release, LCFGng configuration files for that release, and useful LCFGng utility scripts, respectively.

2.3.1 RPM Lists

The RPM lists in the `ng_rpmlist` subdirectory are suitable for LCFGng. The lists are separated into a large number of lists for individual services (usually both server and client lists) and a smaller number of files which collect these services into standard machine types. The machines types are listed in Table 2.1 along with a short description of each. On a typical site only the CE, SE, and MON machine types are deployed. The others correspond to specialized services for virtual organization or for the grid as a whole.

RPM lists appropriate for use with `wget` are available on the web⁴. These are derived from the lists kept in the `edg-release` module.

2.3.2 Configuration Files

In the `ng_source` subdirectory are all of the files defining the configuration for the various different machine types. There is a one-to-one correspondence between the RPM lists and the configuration file

⁴<http://datagrid.in2p3.fr/autobuild//rh7.3/rpmlist/>



for each machine type. Table 2.1 gives the name of the configuration file for each machine type.

Unfortunately, there are some parts of the configuration which are not handled automatically by LCFGng. This guide contains the necessary manual configuration step necessary for each service. There is also a set of release notes kept in the top-level of the edg-release module named `edg-release-notes.txt` which contain the latest information.

2.3.3 Utility Scripts

In the `ng.updaterep` subdirectory are a set of script used to maintain the software cache on the LCFGng server and to ensure that the correct version of the LCFGng components are installed on the server. See Chapter 3 for instructions on how these are used.

3 Installation

3.1 Manual Installation

The manual installation of the EDG software involves just installing the appropriate RPMs for a given machine type. However, manual configuration of the services without the benefit of the LCFG configuration components is extremely difficult. In fact no running site has used this method. In the future, the EDG service configuration will be significantly simplified to allow manual configuration. However for now, using this method is strongly discouraged.

3.2 LCFGng Installation

Using an LCFGng server running on a RedHat 7.3 system is the only supported installation and configuration method at this time. Unfortunately installing an LCFGng server for the first time is a high hurdle but once installed makes the rest of the installation rather easy. Upgrades to newer EDG releases are also straight-forward. (Upgrades of the EDG release are rather frequent because of security fixes for underlying software and for bug fixes in the EDG software itself.)

3.2.1 LCFGng Server Installation

Even if you use fairly standard hardware, the installation of the LCFGng server along with the installation and configuration of the first client takes a day or two. If you have unusual hardware, it may take longer. A cookbook ([html](#)¹, [pdf](#)², [ps](#)³) on server installation describes the process well.

If you have unusual hardware you may wish to install the two kernel packages:

```
kernel-lcfg-boot-2.4.20-20.7.edg2.i386.rpm
kernel-lcfg-boot-preinstall-2.4.20-20.7.edg2.i386.rpm
```

on your server. The boot kernel has nearly all known NIC drivers statically compiled into the kernel. It also contains most of the known SCSI and RAID drivers as modules. To get the modules into the NFS volume used for installation, use the following command:

```
rpm --install --root /opt/local/linux/nginstallroot/7.3/ \
    kernel-lcfg-boot-2.4.20-20.7.edg2.i386.rpm
```

This will do a **chroot** to the installation root before installing the RPM.

3.2.2 Client Installation

Most sites install only machines in one of the predefined configurations. If you do so, then the only files you will need to edit when using LCFGng are the top-level machine files, `site-cfg.h` and possibly the `redhat73-cfg.h` file.

¹<http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/documentation/lcfgng-server73.html>

²<http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/documentation/lcfgng-server73.pdf>

³<http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/documentation/lcfgng-server73.ps>

The `site-cfg.h` file is the first to customize for your site. It contains lots of information about your site. The macros defined there are used to set the appropriate values in the other configuration files. You will only have to edit the `redhat73-cfg.h` file if you have very unusual hardware.

There must be a top-level machine file for each LCFGng client you want to install. Examples of each machine type can be found in the `ng_source/examples` subdirectory of the `edg-release` module. Usually this involves copying the appropriate example to the `/var/obj/conf/server/source/` directory, renaming it to the machine name, and editing it to contain the proper host name.

After creating a new top-level machine file, you must update the software cache with the needed RPMs. The command for updating the cache can be found in the `ng_updaterep` subdirectory of the `edg-release` module. You will need to copy the configuration file in the same directory (`updaterep.conf`) to `/etc`. The command takes a list of all of the top-level machine names so typically is invoked like:

```
updateRPMCache.pl /var/obj/conf/server/source/grid*
```

which will scan all of the machine files (`grid01`, `grid02`, etc. in this case) and download the needed packages from the package repository. The command has some inline help to explain other options.

Also you must ensure that the LCFGng server itself has the correct LCFGng components installed. The script `checkServerRPMs.pl` will do this.

After this it is just a matter of making the machine profile (with `mkxprof`) and triggering an installation from scratch as you did for the first LCFGng client. (Typically setting the PXE install to point to the `nginstallroot` and restarting the machine.) After the machine finishes the installation, you need to do any manual configuration specified for the services on that machine and reboot the machine. (Note: because of a problem with various perl RPMs, the first reboot will give an “updaterpms failed” error. However, the installation will continue normally. Updates do not suffer from this problem.)

3.2.3 Cron-Note

(11) Various cron jobs must be given time to run before anything will work after a fresh install, e.g. for creating grid-mapfiles and downloading the latest CRLs. In order to run these by hand you should check three locations for cronjobs:

- `$EDG_LOCATION/etc/cron`
- `/var/spool/cron/root`
- `/etc/crontab`

3.2.4 Known Problems

- Bug #1902: (inconsistency of EDG-release provided boot kernel and modules)
- Bug #none: sometimes, when running the server installation (`create-nginstallroot-rh73`), the following warnings/errors may be issued:

```
[4/4] installing installroot RPMs...
warning: created %_tmppath directory /var/local/tmp
Cannot create temp file /tmp/fstab.tU4bX1
/var/local/tmp/rpm-tmp.72422: $TMP: ambiguous redirect failed
to add devpts filesystem to /etc/fstab
error: execution of %post scriptlet from dev-3.3-4 failed, exit status 1
```

these can be safely ignored.

- (22) On the first installation from scratch there will be an "updaterpms failed" message. This is a byproduct of adding the 'd' flag to the perl and perl-suidperl rpms to fix a hang in the installation. Despite the error, the machine should continue to install and complete successfully.

3.3 UML-LCFGng Installation

Installing an LCFGng server for the first time is a high hurdle and requires on additional machine. For this reason an example LCFGng server was deployed inside a User Mode Linux (UML). It consists of one image file which can be booted using an executable kernel. Both are provided. This UML-LCFGng is configured to work as an example that only requires basic adjustments by site admins.

UML can be run on any recent Linux Distribution without heavy requirements on the installed kernel. A cookbook⁴ installation guide is available. The image provided contains an LCFGng server installed on RH7.3 using the cookbook method that was mentioned in 3.2. Since this image was developed within the EU CrossGrid project, there are some slight modifications that should be known prior to the installation:

- The cookbook references the GridPortal instead of the Marianne site for the Download Repository, which is essentially a mirror of Marianne and LCG plus the X# Applications.
- the CVS repository is used from GridPortal as well. It contains a slightly different structure in order to hold the configuration of all X# sites. The EDG repository is being mirrored into `common/edg`. This can easily be changed.

The site's configuration files itself are held in `my_site` in order to avoid messy configuration modifications.

⁴<http://gridportal.fzk.de/websites/crossgrid/site-fzk/UML-LCFG.txt>

4 Authentication, Authorization, and Security

This chapter describes some of the details concerning the EDG security framework. System administrators should be familiar with the information here, but apart from a couple of points, the configuration done via LCFG handles all of the necessary details.

The first point is that there is no secure way to distribute host certificates via LCFG. Consequently after installing a host, the host certificate must be copied into place by hand. The host should then be rebooted to ensure that all necessary configuration is done.

The second point is that the site's firewall is outside of the control of LCFG. Most of the EDG services require inbound IP connectivity on specific ports; most also require outbound IP connectivity. (See Section 4.4.)

4.1 Access Policy

Most sites require that users of computing resources agree to the site's usage requirements. This is often enforced by signing a list of those requirements. This technique scales poorly when there is more than a few sites and the users are geographically distributed.

Having no usage policy is clearly not acceptable; therefore, EDG developed a common usage policy in consultation with the partners of the project. Users can sign this policy (via their certificate) by visiting a web page. Sites can require that users have done so by requiring that the user be listed in the "Guidelines" virtual organization.

The policy itself is available on the testbed documentation web page¹. Site administrators should be familiar with its contents.

4.2 Authentication

Cryptographic certificates are used to attest to the identity of an user or machine to the extent specified in the issuing certification authority's (CA) policy documents. Users accessing DataGrid resources must have a valid certificate; similarly, hosts offering services within the testbed must also have one.

The EDG-approved CAs have service areas which cover most of Europe and the United States. (Consult the current list² on the web.) If a user or site is not covered by an existing CA's service area, then one must either negotiate with a CA to extend its service area or start a new CA.

The DataGrid-fr CA is the catch-all CA which issues certificates for users not covered by an existing CA.

4.2.1 Installing User Certificates

To use the Globus security infrastructure you must have your certificate in PEM format. Follow the instructions below if you need to change a P12-formatted certificate into a PEM-formatted certificate. You should then place the two files `usercert.pem` and `userkey.pem` into a `.globus` directory in your home area. The file permissions for the userkey file should be 0700; 0755 is appropriate for the other.

Optionally, you may place your certificate and key in a non-standard location. In this case you must define the two environmental variables `X509_USER_CERT` and `X509_USER_KEY` to point to your certificate and key, respectively.

¹<http://marianne.in2p3.fr/datagrid/documentation>

²<http://marianne.in2p3.fr/datagrid/ca/ca-table-ca.html>

4.2.2 Installing Host Certificates

Host certificate/key pairs should be installed into the directory `/etc/grid-security/`. The host key must be readable only by root (**`chmod 0400 hostkey.pem`**); the host certificate can be world readable (**`chmod 0444 hostcert.pem`**).

These certificates may be installed in non-standard locations by setting the values `X509_USER_CERT` and `X509_USER_KEY` to the appropriate values in the `/etc/sysconfig/edg` and `globus.conf` configuration files.

4.2.3 Changing Certificate Formats

Many of the certificate authorities deliver certificates through a web browser. To use these certificates with Globus, they must be exported from the browser and then reformatted for Globus. Exporting is browser-specific so you will need to follow the help provided with your browser. Once you have extracted the certificate you should have a file with a `p12` extension. This is in the PKCS12 format; you will need to change this to PEM format. If the `edg-utils` package is installed on your machine, simply executing **`pkcs12-extract`** will create appropriate certificate and key files and place them in the standard location. This is a convenience method for the following:

```
openssl pkcs12 -nocerts \  
    -in cert.p12 \  
    -out ~user/.globus/userkey.pem  
  
openssl pkcs12 -clcerts -nokeys  
    -in cert.p12  
    -out ~user/.globus/usercert.pem
```

The first command gives you your private key; this file must be readable only by you. The second command gives your public certificate. The “`~user`” should be replaced by the path to your home area. The `.globus` directory is standard place to put your certificates.

Popular browsers typically use certificates in PKCS12 format. Consequently you will need to modify the format of the PEM certificates used for Globus to use them within a browser. To change a certificate from PEM format into PKCS12 format (on a machine with `edg-utils` installed), just issue the command **`grid-mk-pkcs12`**. Again, this is a convenience method for the following:

```
openssl pkcs12 -export \  
    -out file_name.p12 \  
    -name "my-certificate" \  
    -inkey ~user/.globus/userkey.pem \  
    - in ~user/.globus/usercert.pem
```

where `file_name.p12` is the name of the PKCS12 certificate, and the “`~user`” in the last two lines should be replaced by the path to your home area. You must then import the certificate into your browser.

4.2.4 Updating Certificate Revocation Lists

Having current certificate revocation lists (CRLs) is an extremely important aspect of the security framework. These lists identify certificates which have been revoked because the user no longer uses them, or they have been compromised. The CRLs can be updated with the command **`edg-fetch-crl`**. There is an associated daemon (**`edg-crl-upgraded`**) which can be started automatically to retrieve the CRLs periodically. It can be manipulated like all SysV daemon scripts.

Note: if the CRLs are out-of-date, certificates from the associated CA will not be accepted.

4.2.5 Known limitation on keylength

The JCE (java cryptography extension) as deployed places limits on the sizes of keys. The FNAL CA is using a 4096 bit key which exceeds these limits. Consequently, the FNAL CA cannot be supported by java services with the JDK as distributed. The security group is investigating the legal issues surrounding the use of the unlimited strength JCE policy.

4.3 Authorization

Virtual Organizations (VOs) are used to organize the testbed users into various subgroups and are the basis for grid *authorization*. When a user runs a task, the user's certificate information is compared with a file which is populated by information from the various VOs. "John Doe" may have been added to the Alice VO, in which case the file referred to will have an entry for "John Doe" along with a directive to map his requests onto a local Alice environment. On the other hand, John would not be allowed to run jobs under other environments.

Sites have control over what VOs they accept and over which local account a particular grid user will use.

4.3.1 Virtual Organizations

The current list of virtual organizations³ can be found on the web. If you did not register with a virtual organization when you signed the EDG Usage Guidelines (or wish to change your VO membership), then you must contact the VO manager directly.

If none of the listed virtual organizations is appropriate for you, use the WP6 VO. It is intended as a catch-all for folks who are not members of any of the others.

With the Testbed 2 software, membership in more than one virtual organization is possible. However the burden of specifying which virtual organization falls to the user. The local account is still determined by the site's configuration and mismatches may cause problems when standard unix permissions are used by services.

4.3.2 VO-specific Software

Most of the virtual organizations currently require that some VO-specific software is preinstalled at sites supporting that virtual organization. The availability of VO-specific software on a site is published into the information systems from the `/opt/edg/info/mds/etc/ldif/ce-static.ldif` file by setting one or more `RunTimeEnvironment` attributes.

The list of RPMs can be obtained from the EDG repository.

4.3.3 Local Accounts

Access to a site is controlled through the grid credentials, but once on a site most software uses the local uid and gid of the account. There are several techniques for mapping grid users into local accounts. The default is to use the "pooled" account technique.

³<http://marianne.in2p3.fr/datagrid/vo/vo-table.html>

Individual Accounts

The configuration file allows for three different strategies for creating the local user accounts, each with advantages and disadvantages. The first option is to create a unique local account for every grid user. This allows the environment for each user to be specifically tailored for that user and allows detailed accounting of resource usage through standard mechanisms. The disadvantage is that this involves a lot of maintenance by the system administrator and may involve a large number of accounts being created.

Shared Account

The second option is for all members of a particular virtual organization to be mapped into a shared account. Administratively this is the easiest solution as it usually involves only setting up one account per virtual organization. However, all detailed accounting information is lost, detailed access control is more difficult, and there are possible resource conflicts between multiple users at the same site.

Pooled Accounts

The third option is creating pooled accounts. It is similar to the last option but instead pools of identical accounts are created and at any given time only one user (identified by subject name) is using one account. For example, for the Atlas VO a site may create a pool of accounts atlas001, atlas002, etc. This has the advantages that the accounts are easier to maintain and allow detailed accounting. However, there is a need to specify a policy for local resources when a given user stops using a pooled account. (E.g. how long local files are maintained, will the user get the same account when she/he returns, etc.)

Configuration In addition to the configuration of the `mkgridmap` script (described below), the accounts and a `gridmapdir` must be setup.

To create a pool of accounts, you must setup individual unix user accounts whose names have a common prefix and a numeric suffix. For example, "atlas001", "atlas002", etc. To map users into this pool the prefix must be specified preceded with a dot, i.e. ".atlas".

In addition, a `gridmapdir` must be created; its default location is `/etc/grid-security/gridmapdir`, but may be set to a different location in the `globus.conf` file. An empty file must exist in the `gridmapdir` for each pooled account; the name of the file must match exactly the account name including both prefix and numeric suffix.

The mapping between a subject name and an individual account is based on the time stamps of the account entries in the `gridmapdir` and additional files named according to the URL-encoded subject names of the users.

Note: this mapping is fixed until the subject name entry is deleted. Currently this isn't done automatically and if the account pool is exhausted, users will get the same error as if they were not authorized to use the resource.

One important aspect for using pooled accounts is that the `grid-mapfile` and the `/etc/grid-security/gridmapdir` directory must be shared between all of the nodes in a site. If this is not done, then it is possible that the mapping will be done inconsistently depending on how a given machine is accessed.

Generating Mapfile with `mkgridmap`

The `mkgridmap` script generates a `gridmap` file based on user information in the LDAP servers of various virtual organizations.

The behaviour of the script can be highly customized via a configuration file which is located in `/opt/edg/etc/mkgridmap.conf`. In its simplest form, it simply lists the appropriate virtual organizations, the accounts to map these users to, and an **auth** directive to check that the users have signed the EDG Usage Guidelines.

The following example file (appropriate for a computing element) maps users from the specified virtual organizations to pooled accounts with the given prefix.

```
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=alice,dc=eu-datagrid,dc=org .alice
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=atlas,dc=eu-datagrid,dc=org .atlas
group ldap://grid-vo.nikhef.nl/ou=tblusers,o=cms,dc=eu-datagrid,dc=org .cms
group ldap://grid-vo.nikhef.nl/ou=tblusers,o=lhcb,dc=eu-datagrid,dc=org .lhcb
group ldap://grid-vo.nikhef.nl/ou=tblusers,o=biomedical,dc=eu-datagrid,dc=org .biome
group ldap://grid-vo.nikhef.nl/ou=tblusers,o=earthob,dc=eu-datagrid,dc=org .eo

group ldap://marianne.in2p3.fr/ou=ITeam,o=testbed,dc=eu-datagrid,dc=org .iteam
group ldap://marianne.in2p3.fr/ou=wp6,o=testbed,dc=eu-datagrid,dc=org .wpsix

auth ldap://marianne.in2p3.fr/ou=People,o=testbed,dc=eu-datagrid,dc=org
```

This also checks the generated list of users against those who have signed the EDG Usage Guidelines. An example appropriate for a resource broker

```
group ldap://marianne.in2p3.fr/ou=guidelines,o=testbed,dc=eu-datagrid,dc=org dguser
auth ldap://marianne.in2p3.fr/ou=People,o=testbed,dc=eu-datagrid,dc=org
```

checks only the group of users who have signed the EDG Usage Guidelines and maps them into the user which runs the broker daemons.

4.4 Firewalls

Most of the EDG software requires inbound IP connectivity on various ports to function correctly. Table 4.1 lists those ports used by various parts of the testbed software. Temporary ports used by Globus can be restricted to a particular range. Nearly all services can be configured to run on non-standard ports, if necessary.

The EDG software (and the users) expect outbound IP connectivity from all of the service and worker nodes.

4.5 Miscellaneous Comments

When mixing services on a given machine, be especially careful about what services you run on a machine that grid users can access. For example, MySQL is used by many services and uses only simple passwords for protection. If user jobs are able to run on the same node and access the database, the integrity of the grid service will be compromised. In general direct user access should be disallowed from any node running a grid service.

Service	Ports*	Open To	GRID		Virtual Organization							Site				
			IC	MDS	VOMS	MS-LDAP	PX	RB	BDII	RLS	UI	SE	CE	WN	NM	MON
httpd (apache)	80	world	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓
MySQL	3306	machine	✓	-	✓	-	-	-	-	-	-	-	-	-	-	✓
NFS (Network File System)	2049	site	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ntpd (Network Time Protocol)	123/udp	ntp server(s)	✓	-	✓	-	-	-	-	-	-	-	-	-	-	✓
OpenLDAP	389	world	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓
openssh	22	world	✓	-	✓	-	-	-	-	-	-	-	-	-	-	✓
portmap	111/udp & tcp	site	✓	-	✓	-	-	-	-	-	-	-	-	-	-	✓
Globus & EDG Gatekeepers	2119	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Globus Job Manager	20000-25000†	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GridFTP	2811	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MyProxy	7512	world	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
BDII (LDAP)	2170	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CondorG	7771	world	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
GOUT (LDAP)	2169	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GridFTP (EDG modified)	2811	world	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
Iperf	5001	world	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
LEMON (fab. mon. server)	12409/udp & tcp	site	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LEMON (fab. mon. service)	12411	site	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Logging & Bookkeeping Server	9000, 9001	world	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
locallogger (logd)	9002	world	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
MDS (LDAP)	2135	world	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
Mercury Monitor (local)	3570	gatekeeper	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Mercury Monitor (site)	3570	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Network Server	7772	world	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
pcpd	50500	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RFIO	3147	world	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
R-GMA (tomcat)	8088, 8080, 8443§	world	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
RLS-RLI (tomcat)	8080, 8443§	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RLS-LRC (tomcat)	8080, 8443, 9101-9120†	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RLS-RMC (tomcat)	8080, 8443, 9201-9220†	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RLS-ROS (tomcat)	8080, 8443, 9301-9320†	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RTCS (scheduler input)	6091	machine	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
RTCS (maintenance)	6092	machine	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RTCS Scheduler (Maui)	42559	machine	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SE/apache-ssl	6375	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SE (tomcat)	8080, 8443§	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UDPmon	14233/udp	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VOMS	15000-15020	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VOMS (httpd & mod_ssl)	443	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VOMS (admin. client, tomcat)	8080, 8443§	world	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4.1: Daemon Ports and Machine Types

* All are TCP ports unless otherwise indicated.
 † Selectable range; required with GridFTP, Gatekeeper (no user logins).
 § Secure port, ‡ One secure port per supported VO.

• optional
 ✓ required
 - not required

5 EDG Machine Configuration

Using LCFG, the procedure for the installation and configuration of the various types of machines is straight-forward:

1. Make a copy of the example machine configuration file for the type of machine you are installing.
2. Use LCFG to install the machine.
3. Copy the host certificate and key to the machine.
4. Consult Table 4.1 to determine what services run on that machine. Perform any manual configuration necessary for the services that run on the machine.
5. Reboot.
6. Test.

All machine types can be installed in this way. The only machine type listed separately is the user interface as there is a greater need for people to install this type of machine manually.

5.1 Manual Installation of User Interface

This section assumes that you are trying to install the user interface software on a machine with a preexisting installation of RedHat Linux 7.3. (It has also been done successfully on a RH9 machine.) These instructions assume that you have root access to the machine.

First download the necessary RPMs. Consult the web for the latest list of tags¹ and click on the link corresponding to the version you want. (This guide refers to the EDG 2.0.x series of releases.) You should see a table with machine types on the left and a series of links to various lists of RPMs. Download the UI lists under the columns “ca”, “edg”, “globus”, and “external”. At the top of each file is a **wget** command line which will download the full list.

Next try installing all of the downloaded RPMs:

```
rpm --install *.rpm
```

There will probably be warnings about RPMs which are already installed or conflicts with installed RPMs. The most likely being that “swig” is already installed and that there are more recent versions of “w3c-libwww” and “w3c-libwww-devel” RPMs already installed. You can simply remove them from the list of RPMs to install; the number of RPMs to remove from the list should be small.

Next configure RGMA. As root run the script `/opt/edg/sbin/edg-rgma-config`. Answer the questions which are presented with the defaults. For the “user servlet”, “Schema servlet”, and “registry host”, enter the name of the Information Catalog machine for the testbed. This is `gppui05.gridpp.rl.ac.uk` at the time this guide was written. There will be an error complaining that `prevayler.jar` and that the “edginfo:edginfo” user do not exist. Both errors can be ignored. You should be able to log into a non-root account and use the **edg-rgma** command.

Next configure the workload management client. Go to the directory `/opt/edg/etc` and copy the configuration template:

¹<http://datagrid.in2p3.fr/autobuild//rh7.3/rpmlist/>



```
cp edg_wl_ui_cmd_var.conf.template edg_wl_ui_cmd_var.conf
```

Edit the configuration file to contain the name of the resource broker machine you use for logging and the name of the default VO.

For each supported VO, you must create a directory under `/opt/edg/etc/` with the name of the VO. For example, for the “iteam” VO, create the directory `/opt/edg/etc/iteam`. Then copy the template into place,

```
cp /opt/edg/etc/vo_template/edg_wl_ui.conf /opt/edg/etc/iteam/edg_wl_ui.conf
```

Edit this file with the list of resource brokers you want to use. (Use same nodes for the logging destinations as well.) You should now be able to log in as a non-root user and use the job management commands.

Lastly, to configure the replica manager, copy the template file into place:

```
cd /opt/edg/etc/edg-replica-manager
cp edg-replica-manager.conf.template edg-replica-manager.conf
```

and replace the fields delimited by @’s with appropriate values. The local domain, default CE, and default SE values essentially tell the replica manager where your UI is located in reference to other grid resources. Choose values which make sensible defaults for you. Make the following additional

replacements: INFOSERVICE RGMA This will set the RM to work with RGMA.
 ROS.FAILURE true

6 EDG Service Configuration

6.1 Berkeley Database Information Index (BDII)

Table 6.1: BDII Installation Information

Description:	Cache of MDS information for broker based on OpenLDAP
Supplier:	EDG (WP6, WP4)
Bugzilla Category:	AAA: NONE OF OTHERS
Node type:	BDII
License:	European DataGrid License
LCFG Components:	edg-lcfg-bdiicfg
Config. files:	EDG_LOCATION/etc/bdii.conf
Config. templates:	EDG_LOCATION/etc/bdii.conf.template
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	bdii
SysV directives:	start, stop, status, restart
Daemon username:	edginfo:edginfo
Process name(s):	slapd (typically 5-20)
Process ID (PID) file(s):	EDG_LOCATION_VAR/run/slapd.pid
Lock file(s):	None
Log file(s):	Node
Ports:	2170/tcp
Cron entries:	EDG_LOCATION/etc/cron/bdii-cron (every 10 minutes)
Env. variables:	none
Other daemons contacted:	Top-level MDS
Link(s) to documentation:	Undocumented

See Table 6.1 for summary of installation information.

6.1.1 Post-LCFGng Manual Configuration

None.

6.1.2 Testing and Debugging

To verify that the BDII server is running, try a query:

```
ldapsearch -LLL -H ldap://bdii.example.org:2170
-x -b 'mds-vo-name=local,o=grid'
```

It dump all of the information in the database. As this is a cache of the information in the top-level GIIS, verify that the information from:



```
ldapsearch -LLL -H ldap://top-giis.example.org:2135  
-x -b 'mds-vo-name=local,o=grid'
```

matches that from the query on the BDII.

There is no log file produced by default. If you need access to log information, capture the command used to start the daemon from the BDII init.d script and run it interactively.

6.2 EDG Gatekeeper

See Table 6.2 for summary of installation information.

6.2.1 Post-LCFGng Manual Configuration

- The host certificates must be copied into place manually after the initial installation. The machine should then be rebooted after all manual configuration to ensure that all LCFG components run.
- (1)¹The WP4 resource management software is available, but not currently configured by default. Uncomment the `rtcs` and `maui` configurations in the `ComputingElement-cfg.h` file to start these services.
- (2) Configuration of local resource management system must be done manually. For PBS (the default), the PBS clients are configured automatically via LCFG; the PBS server on the gatekeeper needs some manual configuration. For details on PBS configuration see 12.
- (3) The user accounts must have home area on a shared file system which is mounted on the gatekeeper and all worker nodes. Moreover, each account must have an old and new-style ssh key. An ssh command on each worker node must work to the same worker node without any challenges about the host key. (Jobs fail with "host key validation failed" messages.) If MPI is supported (installed by default), then the ssh must work without challenge to all other worker nodes on the cluster as well. There are objects to do this configuration, but when WNs are added, the cron entries should be run by hand.

Note that if a WN or CE is reinstalled and its ssh identity changes then `/etc/ssh/ssh_known_hosts` must be deleted from all WNs and CEs

The object that sets up the user keys take considerable time to run after a fresh install.

- (16) For LCMAPS you must configure the `groupmapfile` on the gatekeeper manually. Create a file `$EDG_LOCATION/etc/lcmaps/groupmapfile` describing the group (gid) mapping on your site. An example configuration file `groupmapfile.in` is included with the LCMAPS RPM.

6.2.2 Testing and Debugging

Before testing the gatekeeper itself, ensure that the underlying batch system is configured correctly and properly processes jobs. Moreover, verify that a simple `ssh` command executes without challenge on each worker node:

```
ssh worker-1.example.org /bin/echo OK
```

should echo "OK" *without* asking any interactive questions.

Once the batch system is working, try direct submissions with the basic Globus job submission command `globus-job-run`. Test both the fork job manager:

```
globus-job-run gk.example.org/jobmanager-fork /bin/echo OK
```

and your LRMS job manager:

```
globus-job-run gk.example.org/jobmanager-pbs /bin/echo OK
```

substituting "pbs" with the name of your LRMS. Testing both helps separate problems with the gatekeeper and with the underlying batch system.

¹The numbers in brackets refer to the issue mentioned in the release notes

Table 6.2: EDG Gatekeeper Installation Information

Description:	Gatekeeper with security call outs (LCAS and LCMAPS) Local Centre Authorization Service (LCAS) Local Credential Mapping Service (LCMAPS)
Supplier:	EDG (WP4)
Bugzilla Category:	AAA: NONE OF OTHERS
Node type:	CE
License:	European DataGrid License, Globus Toolkit License
LCFG Components:	globuscfg lcas lcmaps
Config. files:	/etc/sysconfig/globus /etc/globus.conf /opt/globus/etc/globus-gatekeeper.conf EDG.LOCATION/etc/lcmaps/groupmapfile (LCMAPS) EDG.LOCATION/etc/lcmaps/lcmaps.db (LCMAPS) EDG.LOCATION/etc/lcmaps/vomapfile (LCMAPS) EDG.LOCATION/etc/lcas/ban_users.db (LCAS) EDG.LOCATION/etc/lcas/lcas.db (LCAS) EDG.LOCATION/etc/lcas/lcas_voms.gacl (LCAS) EDG.LOCATION/etc/lcas/timeslots.db (LCAS)
Config. templates:	EDG.LOCATION/etc/lcmaps/groupmapfile.in (LCMAPS) EDG.LOCATION/etc/lcmaps/lcmaps.db.in (LCMAPS) EDG.LOCATION/etc/lcmaps/vomapfile.in (LCMAPS) EDG.LOCATION/etc/lcas/allowed_users.db.in (LCAS) EDG.LOCATION/etc/lcas/ban_users.db.in (LCAS) EDG.LOCATION/etc/lcas/lcas.db.in (LCAS) EDG.LOCATION/etc/lcas/lcas_voms.gacl.in (LCAS) EDG.LOCATION/etc/lcas/timeslots.db.in (LCAS)
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	globus-gatekeeper
SysV directives:	start, stop, status, restart, reload, condrestart
Daemon username:	root
Process name(s):	edg-gatekeeper (1 process)
Process ID (PID) file(s):	None
Lock file(s):	/var/lock/subsys/globus-gatekeeper
Log file(s):	/var/log/globus-gatekeeper.log
Ports:	2119/tcp
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://www.dutchgrid.nl/DataGrid/wp4/lcas/ (LCAS) http://www.dutchgrid.nl/DataGrid/wp4/lcmaps/ (LCMAPS)

6.3 GOUT

Table 6.3: GOUT Installation Information

Description:	Caches R-GMA information into LDAP server for broker
Supplier:	EDG (WP3)
Bugzilla Category:	R-GMA
Node type:	MON
License:	European DataGrid License
LCFG Components:	edg-lcfg-rgma
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	edg-rgma-gout
SysV directives:	start, stop, status
Daemon username:	edginfo
Process name(s):	slapd
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	None
Ports:	2169/tcp
Cron entries:	None
Env. variables:	None
Other daemons contacted:	R-GMA IC and R-GMA produces of resource information
Link(s) to documentation:	mailto:S.M.Fisher@rl.ac.uk

See Table 6.3 for summary of installation information.

See Section 6.12 for instructions about R-GMA and the MySQL databases running on MON and IC.

6.3.1 Post-LCFGng Manual Configuration

6.3.2 Testing and Debugging

6.4 TCP Throughput Measurement (IPerf)

Table 6.4: IPerf Installation Information

Description:	TCP throughput measurement.
Supplier:	EDG (WP7)
Bugzilla Category:	Network Monitoring
Node type:	Network Monitoring node
License:	European DataGrid License
LCFG Components:	netmon
Config. files:	EDG_LOCATION/etc/edg-iperf/iperf_list.txt Automatically downloaded from central configuration server each night
Config. templates:	none
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	none
SysV directives:	none
Daemon username:	root
Process name(s):	iperf
Process ID (PID) file(s):	none
Lock file(s):	None
Log file(s):	none
Ports:	5001/tcp
Cron entries:	/etc/cron.d/edg-iperf
Env. variables:	none
Other daemons contacted:	none
Link(s) to documentation:	http://ccwp7.in2p3.fr/

See Table 6.4 for summary of installation information.

6.4.1 Post-LCFGng Manual Configuration

6.4.2 Testing and Debugging

6.5 Job Controller (edg-wl-jc)

Table 6.5: Job Controller Installation Information

Description:	Interface between broker daemons and CondorG
Supplier:	EDG (WP1)
Bugzilla Category:	Workload Management System
Node type:	RB
License:	European DataGrid License
LCFG Components:	wlconfig
Config. files:	EDG_LOCATION/edg_wl.conf
Config. templates:	EDG_LOCATION/edg_wl.conf.template
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-jc
SysV directives:	start, stop, status, restart, check
Daemon username:	edguser
Process name(s):	edg-wl-job_cont
Process ID (PID) file(s):	/var/run/edg-wl-job_controller.pid
Lock file(s):	None
Log file(s):	EDG_TMP/jobcontrol/log/events.log
Ports:	??
Cron entries:	EDG_LOCATION/sbin/edg-rgma-publish-service (hourly)
Env. variables:	None
Other daemons contacted:	CondorG master
Link(s) to documentation:	http://www.infn.it/workload-grid/documents.html

See Table 6.5 for summary of installation information.

6.5.1 Post-LCFGng Manual Configuration

6.5.2 Testing and Debugging

6.6 Logging & Bookkeeping Server (edg-wl-lb)

Table 6.6: L&B Server Installation Information

Description:	Logging & bookkeeping server for job state information
Supplier:	EDG (WP1)
Bugzilla Category:	Workload Management System
Node type:	RB
License:	European DataGrid License
LCFG Components:	lbconfig
Config. files:	EDG_LOCATION/etc/edg_wl_query_index.conf
Config. templates:	EDG_LOCATION/etc/edg_wl_query_index.conf.template
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-lbserver
SysV directives:	start, stop, status, restart, proxy
Daemon username:	edguser
Process name(s):	edg-wl-bkserverd (around 10 processes)
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	None
Ports:	9000/tcp, 9001/tcp
Cron entries:	/sbin/service edg-wl-lbserver proxy (hourly)
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://www.infn.it/workload-grid/documents.html

See Table 6.6 for summary of installation information.

6.6.1 Post-LCFGng Manual Configuration

6.6.2 Testing and Debugging

6.7 Fabric Monitoring (LEMON)

Table 6.7: Fabric Monitoring (LEMON) Installation Information

Description:	Fabric monitoring server and sensor agent
Supplier:	EDG (WP4)
Bugzilla Category:	Fabric Monitoring
Node type:	MON (server), All (agent)
License:	European DataGrid License
LCFG Components:	fmonserver fmonagent
Config. files:	EDG_LOCATION/var/etc/edg-fmon-server.conf (server) EDG_LOCATION/var/etc/edg-fmon-agent.conf (agent)
Config. templates:	EDG_LOCATION/etc/edg-fmon-server.conf (server) EDG_LOCATION/etc/edg-fmon-agent.conf (agent)
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	edg_fmon_server (server) edg_fmon_agent (agent)
SysV directives:	start, stop, status, restart, reload
Daemon username:	root
Process name(s):	edg-fmon-server (server) edg-fmon-agent (agent)
Process ID (PID) file(s):	EDG_LOCATION/var/fmon/edg-fmon-server.pid (server) EDG_LOCATION/var/fmon/edg-fmon-agent.pid (agent)
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/fmon/edg-fmon-server.log (server) EDG_LOCATION_VAR/fmon/edg-fmon-agent.log (agent)
Ports:	12409/udp,tcp (server data collection) 12411/tcp (server queries)
Cron entries:	None
Env. variables:	None
Other daemons contacted:	edg-fmon-server (agents contact server)
Link(s) to documentation:	http://cern.ch/lemon

See Table 6.7 for summary of installation information.

6.7.1 Post-LCFGng Manual Configuration

6.7.2 Testing and Debugging

6.8 Local Logger

Table 6.8: Local Logger Installation Information

Description:	Daemon client of logging & bookkeeping server
Supplier:	EDG (WP1)
Bugzilla Category:	Workload Management System
Node type:	CE, RB
License:	European DataGrid License
LCFG Components:	
Config. files:	??
Config. templates:	??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-locallogger
SysV directives:	start, stop, status, restart, proxy
Daemon username:	edguser
Process name(s):	edg-wl-logd (1) edg-wl-interlogd (around 5)
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	None
Ports:	9002/tcp
Cron entries:	/sbin/service edg-wl-locallogger proxy (hourly)
Env. variables:	None
Other daemons contacted:	Logging & Bookkeeping server
Link(s) to documentation:	http://www.infn.it/workload-grid/documents.html

See Table 6.8 for summary of installation information.

6.8.1 Post-LCFGng Manual Configuration

6.8.2 Testing and Debugging

6.9 Local Replica Catalog (LRC)

Table 6.9: LRC Installation Information

Description:	Local replica catalog for, part of the Replica Location Service (RLS)
Supplier:	EDG (WP2)
Bugzilla Category:	Replica Location Service
Node type:	RLS
License:	European DataGrid License
LCFG Components:	lrc
Config. files:	??
Config. templates:	??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-tomcat4
SysV directives:	start, stop
Daemon username:	tomcat
Process name(s):	java (many)
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	/var/tomcat4/logs/catalina.out
Ports:	8080, 8443, 9101-9120
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://cern.ch/edg-wp2/replication/docu/edg-lrc-userguide.pdf

See Table 6.9 for summary of installation information.

Description : One of the two components of the Replica Location Service. The LRC stores GUID to SURL mappings, along with metadata on the physical files and replicas for a given VO

Deployment scenario : One per virtual organization. With the integration of the RLIs in EDG 2.1, the deployment scenario will move to one LRC per SE.

State : Deployed; working stand alone with no interoperation with RLIs

6.9.1 Post-LCFGng Manual Configuration

(12) By default Tomcat only uses a maximum of 64Megs before exiting. Increase this with the lcfg paramter. tomcat4.serverMemMax 1024 to reflect the amount of memory you have on your hardware.

(13) The MySQL tables of the LRC must be set up by hand:

```
$ /opt/edg/sbin/edg-service-mysql-setup.pl \  
  /opt/edg/var/etc/edg-local-replica-catalog
```

LRC/RMC/ROS: Occasional tomcat4 workaround: Tomcat4 as packaged from jpackage in the version we deploy has an intermittent bug where the init.d script loses track of the java processes and cannot shut them down properly; this then causes a variety of problems when it tries to restart, sometime with a half started server. Use the following workaround



```
killall java [as tomcat4]
rm -f /var/lock/subsys/tomcat4
service tomcat4 start
```

if the service starts to misbehave.

6.9.2 Testing and Debugging

6.9.3 Known Bugs

- Enforcement of file syntax: Registration of mappings in the catalogs via the RM enforces certain file name formats. This is not enforced via the LRC catalog API. How much of the catalog APIs should be exposed is still an open question. Bug #1007
- Some SQL scripts are deployed to `$EDG_LOCATION/etc/` by the configure scripts. They should be in `$EDG_LOCATION_VAR/etc`. Bug # 1008
- Configuration script depends on R-GMA StreamProducer.props file to publish initial information on service installation. Bug #1036
- No versioning of SQL schemas. Bug #1359 addMapping method throws null pointer after a tomcat restart when the LRC is publishing to RLIs. Bug #1866

6.10 Network Server (edg-wl-ns)(MON??)

Table 6.10: Network Server Installation Information

Description:	Short description.
Supplier:	EDG (WP1)
Bugzilla Category:	Bugzilla Name
Node type:	RB
License:	European DataGrid License
LCFG Components:	mylcfgconfig
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-ns
SysV directives:	Start, Stop, Status, Restart, Proxy
Daemon username:	edguser
Process name(s):	funkydaemon
Process ID (PID) file(s):	/var/run/funkydaemon.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/log/funkydaemon.log
Ports:	none
Cron entries:	EDG_LOCATION/etc/cron/funky-cron (hourly)
Env. variables:	none
Other daemons contacted:	nonfunkydaemon
Link(s) to documentation:	http://www.edg.org/funky.pdf

See Table 6.10 for summary of installation information.

6.10.1 Post-LCFGng Manual Configuration

- (14) The problem is – as it seems – that it differs if you install from scratch or you make an upgrade of an existing installation. In case of upgrade, it seems there is nothing to do, the links to the existing props files are created in the `$EDG_LOCATION_VAR/edg-rgma` directory. In case of install from scratch the new files are created in `$EDG_LOCATION_VAR/edg-rgma` and you must ensure that the `StreamProducer.props` and `XMLConverter.props` are also in the `$EDG_LOCATION/etc/edg-rgma` directory. It is enough to create two following links and then restart the `edg-netmon2rgmad` daemon.

```
ln -s $EDG_LOCATION_VAR/edg-rgma/StreamProducer.props $EDG_LOCATION/etc/edg-rgma/  
ln -s $EDG_LOCATION_VAR/edg-rgma/XMLConverter.props $EDG_LOCATION/etc/edg-rgma/  
/etc/rc.d/init.d/edg-netmon2rgmad restart
```

Delete the first line in the `/etc/cron.d/edg-iperf` file. The correct `edg-iperf` file:

```
29,59 * * * * root $EDG_LOCATION/sbin/edg-iperf-check >/dev/null 2>\&1  
0 3 * * * root $EDG_LOCATION/sbin/edg-iperf-get_config_file.csh >/dev/null 2>\&1
```

* Delete the first line in the `/etc/cron.d/edg-udpmon` file. The correct `edg-udpmon` file:

```
2 0 * * * root $EDG_LOCATION/sbin/edg-udpmon-bw_resp
    >> $EDG_LOCATION_VAR/edg-udpmon/edg-udpmon-bw_resp_log-out 2>\&1
0 3 * * * root $EDG_LOCATION/sbin/edg-udpmon-get_config_file.csh >/dev/null 2>\&1
```

- (15) One of the parameters in the SE/CE configuration is the associated network monitoring node name. The definition of this node is essential for correct functionality of the network cost calculation.

To choose the associated network monitoring node, WP7 provides following guidelines: 1) FQDN of the real network monitoring node located on the same site. 2) In case that site has no local network monitoring node, the associated network monitoring node name should be defined as "virtualnm.<domain.name>".

- (17) To turn on logging for MON and IC boxes, do the following copy:

```
cp $EDG_LOCATION/etc/edg-rgma/log4j.props \
   /var/tomcat4/webapps/R-GMA/WEB-INF/classes/log4j.properties
```

and restart the services.

6.10.2 Testing and Debugging

6.11 Probes Coordination Protocol Daemon (PCPD)

Table 6.11: PCPD Installation Information

Description:	Probes Coordination Protocol daemon is used for distributed scheduling of the network monitoring activities to avoid concurrent intrusive network measurements.
Supplier:	EDG (WP7)
Bugzilla Category:	Network Monitoring
Node type:	Network Monitoring node
License:	European DataGrid License
LCFG Components:	none
Config. files:	none
Config. templates:	none
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	edg-pcpd
SysV directives:	start, stop, status, restart
Daemon username:	root
Process name(s):	edg-pcpd
Process ID (PID) file(s):	EDG_LOCATION_VAR/run/edg-pcpd.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/logs/edg-pcpd.log
Ports:	50500/tcp
Cron entries:	none
Env. variables:	none
Other daemons contacted:	Other edg-pcpd daemons
Link(s) to documentation:	http://ccwp7.in2p3.fr/

See Table 6.11 for summary of installation information.

6.11.1 Post-LCFGng Manual Configuration

6.11.2 Testing and Debugging

6.12 Relational Grid Monitoring Architecture (R-GMA)

Table 6.12: R-GMA Installation Information

Description:	Short description.
Supplier:	EDG (WP3)
Bugzilla Category:	Bugzilla Name
Node type:	RB
License:	European DataGrid License
LCFG Components:	mylcfgconfig
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-ns
SysV directives:	Start, Stop, Status, Restart, Proxy
Daemon username:	edguser
Process name(s):	funkydaemon
Process ID (PID) file(s):	/var/run/funkydaemon.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/log/funkydaemon.log
Ports:	none
Cron entries:	EDG_LOCATION/etc/cron/funky-cron (hourly)
Env. variables:	none
Other daemons contacted:	nonfunkydaemon
Link(s) to documentation:	mailto:S.M.Fisher@rl.ac.uk

See Table 6.12 for summary of installation information.

6.12.1 Post-LCFGng Manual Configuration

- Don't start the wrong tomcat version
- The hosts-deny.xml uses regular expressions not shell pattern matching expressions. i.e. . means any character, * means 0 or more of the preceding character
- Be careful about proxy caches. If you are accessing an R-GMA servlet via a proxy cache then make sure that the proxy cache has access in the hosts-deny.xml file.
- If you get error messages about edg_rgma_python or edg_rgma_perl, do check that you have the correct versions of Swig and Xerces-c (swig-1.3.19-1, xerces-c-1.7.0-RH7.3)
- Do run edg-rgma-check and see if the output looks sensible
- (7) MySQL tables for R-GMA must be set up manually on both all information catalogue and the monitoring box that are running GOUT.

```
mysql -u root -p < /opt/edg/var/edg-rgma/rgma-db-setup.sql
```

you may also want to set up the root password for mysql to be non blank.

For monitoring boxes that do not run gout you do not need to set up tables.

Generally for MySQL you should set a root password as well.

```
$ mysql -u root - mysql -c "update user set password=password('yourpassword');"
$ mysqladmin -u root -p flush-privileges
```

An example implementation can be found inside this² script.

- (12) By default Tomcat only uses a maximum of 64Megs before exiting. Increase this with the `lcfg` paramter. `tomcat4.serverMemMax 1024` to reflect the amount of memory you have on your hardware.
- (18) Information Catalog (IC): Setup the ACL for the IC to prevent rogue sites from registering with the IC. Edit the file `$EDG_LOCATION_VAR/edg-rgma/access-control-list.xml` to contain an ACL like the following:

```
<access-list>
  <hosts-allow>
    <host servlet="*">.*\.gridpp\.rl\.ac\.uk$</host>
    <host servlet="*">.*\.nikhef\.nl$</host>
  </hosts-allow>

  <hosts-deny>
    <host servlet="RegistryServlet">.*</host>
    <host servlet="SchemaServlet">.*</host>
  </hosts-deny>
</access-list>
```

This will allow everyone to use the browser servlet, but deny registry and schema entries to everyone except NIKHEF and RAL. The values of the `host` clauses are regular expressions (so things like `'` and `*` have special meanings).

- (23) If you're running any secure services in tomcat, you'll need to copy the machine's host certificate and key to separate files and change the ownership of the copies to be that of the user running the tomcat servlet container.

6.12.2 Testing and Debugging

- If there is a proxy cache between you and the IC, then to make the command line tool work you need to set up the environment variable `http_proxy` to point to it.
- If you get messages about `edg_rgma_perl` or `edg_rgma_python` then check that `$PERLLIB` or `$PYTHONPATH` are set.

²http://gridportal.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp4/sites/fzk/other/tb_postinst_scripts/extra-stuff

6.13 Replica Manager (RM)

Description : Client tool that integrates all the grid data management components and provides a single entry point to all other grid services and users.

Defaults to use RGMA as info service. Consequently it expects the RGMA java client RPM to be there. Since this is not used by LCG, we do not put it in as a dependency in the RM RPM.

Deployment scenario: Installation on any UI or WN.

State : Deployed, working. The RM client tool can work with R-GMA, MDS and static configuration files to obtain information about grid services.

Discrepancies with documentation : deleteFile is incorrectly documented, it does not delete entries from the RMC. Bug #1077

6.13.1 Post-LCFGng Manual Configuration

- (20) Replica Manager: Confirm that the file

```
$EDG_LCOATION/etc/edg-replica-manager/edg-replica-manager.conf
```

doesn't exist or delete it if it does. This is created by an old LCFG object and interferes with the current correct configuration file. Once deleted, it won't reappear.

6.13.2 Testing and Debugging

6.13.3 Known Bugs

- Error messages need improvement. Bug #1413
- Double copies when using copyAndRegisterFile. Bug #1417,1756
- copyAndRegister leaves file if registration stage fails. Bug #1676.
- Insecure flag is ignored. Bug #1680
- edg-rm does not give access to metadata. Bug #1685
- No wildcard operations with edg-rm. Bug #1687
- Replica manager does not use info in ServiceStatus table to determine if a service is up. Bug #1690
- Error codes need improvement. Bug #1719
- Performance is very poor, order of magnitude worse than we should be. Bug #1805
- Core-dumps for url-copy with URIs where no GridFTP server is running. Bug #1833
- RM has errors working with ROS. Bug #1838,1769
- Errors catching SE exceptions. Bug #1865.

If you find a bug, please do:



```
cp $EDG_LOCATION/edg-replica-manager/log4j.properties $HOME/log4j.properites
export EDG_RM_LOGPROPERTIES=$HOME/log4j.properties
```

[Edit] \ \$HOME/log4j.properties
and change line 20 from:

```
log4j.logger.org.edg.data=OFF
```

to

```
log4j.logger.org.edg.data=DEBUG
```

rerun the command and send the logfile:

```
/tmp/edg-replica-manager-<username>.log
```

in with the bug report.



6.14 Replica Location Index (RLI)

Description : One of the two components of the Replica Location Service. It is an index that maintains LRC to GUID mappings for all LRCs in a VO that are configured to publish information to it.

Deployment scenario : Many per VO. Deployment according to the needs of a VO State:

State : Not deployed in EDG 2.0. Will be integrated into EDG 2.1.

6.14.1 Post-LCFGng Manual Configuration

6.14.2 Testing and Debugging

6.15 Replica Metadata Catalog (RMC)

Description : A catalog that stores LFN (alias) to GUID mappings along with metadata on the GUID for all files in a given VO.

Deployment scenario : One per virtual organization.

State : Deployed; working.

Table 6.13: RMC Installation Information

Description:	Local Replica Catalogue, part of the Replica Location Service (RLS).
Supplier:	EDG (WP2)
Bugzilla Category:	Replica Location Service
Node type:	RLS
License:	European DataGrid License
LCFG Components:	edg-lcfg-rmc
Config. files:	??
Config. templates:	??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-tomcat4
SysV directives:	start, stop
Daemon username:	tomcat
Process name(s):	java (many)
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	/var/tomcat4/logs/catalina.out
Ports:	8080, 8443, 9201-9220
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://cern.ch/edg-wp2/replication/docu/edg-rmc-userguide.pdf

See Table 6.13 for summary of installation information.

6.15.1 Post-LCFGng Manual Configuration

MySQL scripts need to be run after an LCFG based install to create the database schema. One can either recreate entirely the database, which will remove all the entries, or upgrade the schema only, which will maintain all existing entries.

(13) The MySQL tables of the RMC must be set up by hand:

```
$ /opt/edg/sbin/edg-service-mysql-setup.pl \  
  /opt/edg/var/etc/edg-replica-metadata-catalog
```

LRC/RMC/ROS: Occasional tomcat4 workaround: Tomcat4 as packaged from jpackage in the version we deploy has an intermittent bug where the init.d script loses track of the java processes and cannot shut them down properly; this then causes a variety of problems when it tries to restart, sometime with a half started server. Use the following workaround



```
killall java [as tomcat4]
rm -f /var/lock/subsys/tomcat4
service tomcat4 start
```

if the service starts to misbehave.

6.15.2 Testing and Debugging

6.15.3 Known Bugs

- Returned error codes need to be improved. Bug # 907.
- GUID attributes cannot be created unless a LFN is created, i.e a fake alias. Bug # 965.
- No versioning of SQL schemas. Bug # 1359.
- No deleteMapping method. removeAlias is not aligned with the functionality of deleteMapping in the LRC API. Will be fixed in release 2.1. Bug # 1590
- View all open RMC bugs

6.16 Replica Optimization Service (ROS)

Description : A service that optimizes access to data in the grid.

Deployment scenario : As many as needed, stateless service.

State : Deployed on EDG MON nodes, working. Only the EDG network cost function is currently taken into account in the optimization algorithm. In principle the SE costs algorithm should be integrated as well. The information returned by the SE cost function at the moment is not a useful metric.

Table 6.14: ROS Installation Information

Description:	Replica Optimisation Service: a service responsible for coordinating network queries, allowing optimum replica selection.
Supplier:	EDG (WP2)
Bugzilla Category:	Replica Optimization Service
Node type:	MON
License:	European DataGrid License
LCFG Components:	ros
Config. files:	??
Config. templates:	??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-tomcat4
SysV directives:	start, stop
Daemon username:	tomcat
Process name(s):	java (many)
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	/var/tomcat4/logs/catalina.out
Ports:	8080, 8443, 9101-9120
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://cern.ch/edg-wp2/replication/docu/edg-ros-userguide.pdf

See Table 6.14 for summary of installation information.

6.16.1 Post-LCFGng Manual Configuration

LRC/RMC/ROS: Occasional tomcat4 workaround: Tomcat4 as packaged from jpackage in the version we deploy has an intermittent bug where the init.d script loses track of the java processes and cannot shut them down properly; this then causes a variety of problems when it tries to restart, sometime with a half started server. Use the following workaround

```
killall java [as tomcat4]
rm -f /var/lock/subsys/tomcat4
service tomcat4 start
```

if the service starts to misbehave.



6.16.2 Testing and Debugging

6.16.3 Known Bugs

- getBestFile only considers 1 close SE, case of > 1 close SE is ignored. Bug # 1841
- Application server logging is hardwired. Bug # 933

6.17 Runtime Control System (RTCS)

Table 6.15: RTCS Installation Information

Description:	RunTime Control System. Abstraction layer between PBS and Maui scheduler.
Supplier:	EDG (WP4)
Bugzilla Category:	RMS
Node type:	CE
License:	European DataGrid License
LCFG Components:	rtcs
Config. files:	EDG_LOCATION/etc/rtcs.conf
Config. templates:	None
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	edg-rtcs.d
SysV directives:	start, stop, status, restart, reload
Daemon username:	rtcs
Process name(s):	edg-rtcs
Process ID (PID) file(s):	EDG_LOCATION_VAR/rtcs/rtcs.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/rtcs/rtcs.log
Ports:	6091 (scheduler input) 6092 (maintenance) 42559 (Maui scheduler)
Cron entries:	None
Env. variables:	None
Other daemons contacted:	maui
Link(s) to documentation:	EDG_LOCATION/share/doc/edg-rtcs-*/rms.ps http://datagrid.in2p3.fr/cgi-bin/cvsweb/ checkout /fabric_mgt/resource_mgt/rms/

See Table 6.15 for summary of installation information.

6.17.1 Post-LCFGng Manual Configuration

The WP4 resource management software is installed but not currently configured by default. Uncomment the `rtcs` and `maui` configurations in the `ComputingElement-cfg.h` file to start these services.

6.17.2 Testing and Debugging

6.18 Storage Element Web Service (SEWS)

Table 6.16: Storage Element Web Service Installation Information

Description:	SE with web service interface
Supplier:	EDG (WP2)
Bugzilla Category:	Storage Element
Node type:	SE
License:	European DataGrid License
LCFG Components:	se sewebservice
Config. files:	EDG_LOCATION/etc/se/se-paths.conf EDG_LOCATION/etc/se/se-vos.conf EDG_LOCATION/sbin/edg-se-configure-all.sh
Config. templates:	None
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	??
SysV directives:	??
Daemon username:	edguser
Process name(s):	java
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	/var/log/tomcat4/edg-se-webservice_log
Ports:	tomcat ports
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://web01.esc.rl.ac.uk/projects/DataGrid/wp5/documentation.html http://web01.esc.rl.ac.uk/projects/DataGrid/wp5/documentation/admin/configuring_the... http://web01.esc.rl.ac.uk/projects/DataGrid/wp5/documentation/admin/configuring_the... http://web01.esc.rl.ac.uk/projects/DataGrid/wp5/documentation/user/06_03/faq.html

See Table 6.16 for summary of installation information.

6.18.1 Post-LCFGng Manual Configuration

- Do you need to access a mass storage system? In that case, ensure that the libraries you need are installed (libshift, flfsys, etc).
- Think about which file transfer protocols you need. For GridFTP you need the Globus wu-ftpd. For RFIO, you will need rfiod. Unless you have CASTOR RFIO running on port 5001 (or anything other than the IANA standard), you should probably run rfiod on the standard port, 3147.
- The replica manager calls the SE in **insecure** mode. This means that everybody gets mapped into the 'wpsix' directory. Files created in insecure mode can thus not be accessed in secure mode unless you happen to be a member of 'wpsix'.
If you need to access files via the EDG RM, do not use secure mode.
- (19) Storage Element (SE): The LCFG object for the storage element assumes it is running in secure mode and creates files under the SE root (usually */flatfiles*) like:

```
/flatfiles/dzero  
/flatfiles/lhcb  
/flatfiles/iteam  
...
```

However for EDG 2.0.x, the SE is not running in secure mode, so the corresponding directories must be created under the wpsix directory:

```
/flatfiles/wpsix  
/flatfiles/wpsix/dzero  
/flatfiles/wpsix/lhcb  
/flatfiles/wpsix/iteam  
...
```

If this isn't done, then replication with the replica manager will fail because the SE will report a file size of zero for the replicated file (because it looks in the wrong place).

See also Known Bugs Below

6.18.2 Testing and Debugging

- Check that Tomcat is running.
- Use the `edg-se-webservice` clients to test whether the SE is working
- (6) On the EDG-SE the `grid-mapfile` must be in the default location `/etc/grid-security/grid-mapfile`. It is okay to enter a symlink here to any other location.

6.18.3 Known Bugs

- Since the SE in TB2.0 is not running in secure mode you will run into bug #1334:
GridFTP will map you to a pool account user and will write the file into the SE's disk cache using that id. The SE will think you are John Gordon.
Consequently, you cannot delete re-cached files from the disk cache using `edg-gridftp-rm`, unless you happen to be John Gordon (and even he can only do it until his UKHEP cert expires).
- Also note that the `--vo` switch to the `edg-se-webservice` command does not get propagated to the SE – and it shouldn't. The SE gets the VO information from the `gridmapfile` (pre-VOMS), and from the VOMS credential (when the SE is VOMSified).
- Do not rely on `getMetadata` for this purpose - bug #1333 has not been fixed in TB2.0.
- Delete is known not to work in 2.0. (Fixed in 2.1).

6.19 UDPMon

Table 6.17: UDPMon Installation Information

Description:	UDP throughput measurement.
Supplier:	EDG (WP7)
Bugzilla Category:	Network Monitoring
Node type:	Network Monitoring node
License:	European DataGrid License
LCFG Components:	netmon
Config. files:	none
Config. templates:	none
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	none
SysV directives:	none
Daemon username:	root
Process name(s):	edg-udpmon-bw_resp
Process ID (PID) file(s):	none
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/edg-udpmon/edg-udpmon-bw_resp_log-out
Ports:	14233/udp
Cron entries:	edg-udpmon
Env. variables:	none
Other daemons contacted:	none
Link(s) to documentation:	http://ccwp7.in2p3.fr/

See Table 6.17 for summary of installation information.

6.19.1 Post-LCFGng Manual Configuration

6.19.2 Testing and Debugging

6.20 Virtual Organization Membership Service (VOMS)

Table 6.18: VOMS Installation Information

Description:	Short description.
Supplier:	EDG (WP2, WP6)
Bugzilla Category:	Bugzilla Name
Node type:	RB
License:	European DataGrid License
LCFG Components:	mylcfgconfig
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-ns
SysV directives:	Start, Stop, Status, Restart, Proxy
Daemon username:	edguser
Process name(s):	funkydaemon
Process ID (PID) file(s):	/var/run/funkydaemon.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/log/funkydaemon.log
Ports:	none
Cron entries:	EDG_LOCATION/etc/cron/funky-cron (hourly)
Env. variables:	none
Other daemons contacted:	nonfunkydaemon
Link(s) to documentation:	mailto:Akos.Frohner@cern.ch

See Table 6.18 for summary of installation information.

6.20.1 Post-LCFGng Manual Configuration

6.20.2 Testing and Debugging

Building the package from the source:

```
ant dist
```

Installing it to a directory:

```
ant install -Dprefix=/path/to/dir
```

Building the RPM packages:

```
ant release
```

The RPMs will be under release/rpm/RPMS/noarch.

Configuring the packages (obligatory parameters):

```
<install_dir or dist>/sbin/edg-voms-admin-configure install \  
--vo=<VO_name> --port=<port_number> --dbapwd=<password>
```

To configure the service a MySQL database server has to be running, because the configuration scripts sets up a custom database.

For more details read the manual page of this script.

Run the service:

```
$EDG_LOCATION/etc/init.d/edg-voms-admin start
```

This context block will be automatically copied to the tomcat4/webapps and tomcat4/webapps-secure directories.

The service needs a proper service certificate to allow SOAP or web clients to connect.

Stop the service:

```
$EDG_LOCATION/etc/init.d/edg-voms-admin stop
```

Remove the context blocks from the above mentioned tomcat4 directories.

Remove the service:

```
<install_dir or dist>/sbin/edg-voms-admin-configure remove \  
--vo=<VO_name> --port=<port_number> --dbapwd=<password>
```

One has to call it with the same options as at installation time.

Testing the components:

1. Test if Tomcat is running with a web browser:
`http://localhost:<port-number>/`
2. Test if the edg-voms-admin war file is installed:
`http://localhost:<port-number>/edg-voms-admin/<VO-name>/index.html`
3. Test if the axis components were installed succesfully:
`http://localhost:<port-number>/edg-voms-admin/<VO-name>/happyaxis.jsp`
4. Test if the edg-voms-admin components are properly configured:
`http://localhost:<port-number>/edg-voms-admin/<VO-name>/happyvoms.jsp`
5. Try a test method:
`http://localhost:<port-number>/edg-voms-admin/<VO-name>/services/VOMSCompatibility?
method=getGridmapUsers`
6. Try the admin interface:
`http://localhost:<port-number>/edg-voms-admin/<VO-name>/webui/Admin`

Assign a new administrator to the VO:

If the user's certificate is present in a usercert.pem file, then

```
edg-voms-admin --url=... \  
  create-user usercert.pem \  
  assign-role VO VO-Admin usercert.pem
```



If the service runs on the localhost with automatic local admin setting and the VO's alias is "fred" with VO name "Fred", and the administrator wants to grant himself this permission:

```
edg-voms-admin --url=http://localhost:8080/edg-voms-admin/fred \  
  create-user usercert.pem \  
  assign-role VO VO-Admin usercert.pem
```

or just simply by issuing

```
edg-voms-admin-local fred --add-admin
```

Assign a gridmap file query host to the service:

If the user's certificate is present in a hostcert.pem file, then

```
edg-voms-admin --url=... \  
  add-acl-entry <VO_name> allow list hostcert.pem
```

If the service runs on the localhost with automatic local admin setting and the VO's alias is "fred" with VO name "Fred", and the administrator wants to grant his host this permission:

```
edg-voms-admin --url=http://localhost:8080/edg-voms-admin/fred \  
  add-acl-entry VO allow list /etc/grid-security/hostcert.pem
```

or just simply by issuing

```
edg-voms-admin-local fred --add-host
```

Upgrading the database

If you have an already existing database, then you can insert the new administrative records by issuing the following command:

```
edg-voms-admin-local fred --upgrade-db
```

It may emit error messages, but it is safe to ignore them. It only means, that the database was already up-to-date.

Updating the CA table

The CA table is automatically updated by the service at startup and after a certain period. See the voms.cafiles and voms.cafiles.period properties in the voms.service.properties file!

edg-java-security tips

If you have already installed edg-java-security and edg-java-security-tomcat (starting from release 1.5.3) packages, but the administrative interface still doesn't work, then there are a couple of things to check out:

- /var/tomcat/server/lib/edg-java-security-trustmanager.jar If this jar file is not installed yet, then please run

```
$EDG_LOCATION/sbin/edg-java-security-post-install.sh
```




- `$EDG_LOCATION_VAR/etc/server.xml.d/insecure-service.xml`
`$EDG_LOCATION_VAR/etc/server.xml.d/secure-service.xml` If this xml file is not installed yet, then please run

```
$EDG_LOCATION/sbin/edg-java-security-tomcat-configure
```

- `/etc/init.d/edg-tomcat4` If this file is not at this position, then it is probably the still the default Tomcat service, which is running. Please execute the following commands:

```
ln -s $EDG_LOCATION/etc/init.d/edg-tomcat4 /etc/init.d/edg-tomcat4
ln -s $EDG_LOCATION/etc/init.d/edg-voms-admin /etc/init.d/edg-voms-admin
chkconfig --del tomcat4
chkconfig --add edg-tomcat4
chkconfig --add edg-voms-admin
```

From this point you can start the service by

```
service edg-voms-admin start
service edg-tomcat4 start
```

6.21 Workload Manager (edg-wl-wm)(RB??)

Table 6.19: Workload Manager Installation Information

Description:	Workload manager daemon for broker
Supplier:	EDG (WP1)
Bugzilla Category:	Workload Management System
Node type:	RB
License:	European DataGrid License
LCFG Components:	wlconfig
Config. files:	EDG_LOCATION/etc/edg_wl.conf
Config. templates:	EDG_LOCATION/etc/edg_wl.conf.template
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-wm
SysV directives:	start, stop, status, restart
Daemon username:	edguser
Process name(s):	edg-wl-workload (1)
Process ID (PID) file(s):	EDG_TMP/edg-wl-workload_manager.pid
Lock file(s):	None
Log file(s):	EDG_TMP/workload_manager/log/events.log
Ports:	None
Cron entries:	None
Env. variables:	None
Other daemons contacted:	job controller, logging & bookkeeping server
Link(s) to documentation:	http://www.infn.it/workload-grid/documents.html

See Table 6.19 for summary of installation information.

6.21.1 Post-LCFGng Manual Configuration

- (10) On the RB machine the MySQL tables must be set up.

```
# mysql -u root < /opt/edg/etc/edg-wll-db-setup.sql
# mysqladmin -u root flush-privileges
```

- (21) The following code should be added to the beginning of the start methods in the edg-wl-jc and edg-wl-lm init.d scripts. If not added (and the RB's site has a firewall), all job submissions to sites outside of the firewall will fail.

```
##### Standard EDG Globus configuration #####

GLOBUS_SYSCONFIG=${GLOBUS_SYSCONFIG:-/etc/sysconfig/globus}
if [ -f "$GLOBUS_SYSCONFIG" ]; then
    . "$GLOBUS_SYSCONFIG"
fi

GLOBUS_LOCATION=${GLOBUS_LOCATION:-/opt/globus}
if [ ! -d "$GLOBUS_LOCATION" ]; then
```



```
    echo "GLOBUS_LOCATION not found"
    exit 1
fi
export GLOBUS_LOCATION

if [ -n "$GLOBUS_TCP_PORT_RANGE" ]; then
    export GLOBUS_TCP_PORT_RANGE
fi

if [ -n "$GLOBUS_UDP_PORT_RANGE" ]; then
    export GLOBUS_UDP_PORT_RANGE
fi
##### END Standard EDG Globus configuration #####
```

6.21.2 Testing and Debugging

7 Globus Service Configuration

7.1 GridFTP

Table 7.1: GridFTP Installation Information

Description:	Globus GridFTP server.
Supplier:	Globus
Bugzilla Category:	VDT
Node type:	SE, CE
License:	Globus License
LCFG Components:	globuscfg
Config. files:	/etc/sysconfig/globus /etc/globus.conf
Config. templates:	None
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	globus-gridftp
SysV directives:	start, stop, restart, reload, condrestart, status
Daemon username:	root
Process name(s):	in.ftpd
Process ID (PID) file(s):	/var/run/ftpd.pid
Lock file(s):	/var/lock/subsys/globus-gridftp
Log file(s):	/var/log/globus-gridftp.log
Ports:	2811
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://www.globus.org/

See Table 7.1 for summary of installation information.

7.1.1 Post-LCFGng Manual Configuration

None.

7.1.2 Testing and Debugging

To verify that the server is functioning try the command:

```
edg-gridftp-ls gsiftp://gridftp.example.org/tmp
```

which should list the contents of the /tmp directory on the server.

For a more detailed test use:



```
edg-testbed-test BaseTest::GridFTP gridftp.example.org
```

which is part of the standard test suite. You can check third-party transfers with this test by specifying another GridFTP server on the command line.

7.2 Monitoring and Discovery Service (MDS)

Table 7.2: MDS Installation Information

Description:	Monitoring and Discovery Service (MDS)
Supplier:	Globus
Bugzilla Category:	VDT
Node type:	CE, SE
License:	Globus License
LCFG Components:	globuscfg
Config. files:	/etc/sysconfig/globus /etc/globus.conf
Config. templates:	None
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	globus-mds
SysV directives:	start, stop, restart, reload, condrestart, status
Daemon username:	edginfo
Process name(s):	slapd
Process ID (PID) file(s):	/var/run/globus-mds.pid
Lock file(s):	None
Log file(s):	None
Ports:	2135
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://www.globus.org/

See Table 7.2 for summary of installation information.

7.2.1 Post-LCFGng Manual Configuration

7.2.2 Testing and Debugging

- First you should test, if your local MDS daemon `slapd` works properly. Do this by running similar to this one:

```
ldapsearch -x -H ldap://<YOUR_CE_HOST>:2135/ -b \
'Mds-vo-name=<WHATEVER_YOU_PROVIDED_AS_SITE_GIIS,o=Grid' | grep "dn: .eId"
```

This should give output similar to this:

```
[marcus@ui001:~]$ ldapsearch -x -H ldap://ce010:2135/ -b \
'Mds-vo-name=fzkpro,o=Grid' | grep "dn: .eId"
dn: seId=se010.fzk.de,Mds-Vo-name=fzkpro,o=grid
dn: ceId=ce010.fzk.de:2119/jobmanager-pbs-workq,hn=ce010.fzk.de,Mds-Vo-name=fz
dn: ceId=ce010.fzk.de:2119/jobmanager-pbs-short,hn=ce010.fzk.de,Mds-Vo-name=fz
dn: ceId=ce010.fzk.de:2119/jobmanager-pbs-medium,hn=ce010.fzk.de,Mds-Vo-name=f
dn: ceId=ce010.fzk.de:2119/jobmanager-pbs-long,hn=ce010.fzk.de,Mds-Vo-name=fzk
```

```
ldapsearch -x -H ldap://ce010:2135/ -b 'Mds-vo-name=fzkpro,o=Grid' — grep "dn: .eId"
```

8 Third-party Service Configuration

8.1 CondorG

Table 8.1: CondorG Installation Information

Description:	Short description.
Supplier:	EDG (WPx)
Bugzilla Category:	Bugzilla Name
Node type:	RB
License:	European DataGrid License
LCFG Components:	mylcfgconfig
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-ns
SysV directives:	Start, Stop, Status, Restart, Proxy
Daemon username:	edguser
Process name(s):	funkydaemon
Process ID (PID) file(s):	/var/run/funkydaemon.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/log/funkydaemon.log
Ports:	none
Cron entries:	EDG_LOCATION/etc/cron/funky-cron (hourly)
Env. variables:	none
Other daemons contacted:	nonfunkydaemon
Link(s) to documentation:	mailto:condor-admin@cs.wisc.edu

See Table 8.1 for summary of installation information.

8.1.1 Post-LCFGng Manual Configuration

8.1.2 Testing and Debugging

8.2 Mercury

Table 8.2: Mercury Installation Information

Description:	Short description.
Supplier:	EDG (WP3)
Bugzilla Category:	Bugzilla Name
Node type:	RB
License:	European DataGrid License
LCFG Components:	mylcfgconfig
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-ns
SysV directives:	Start, Stop, Status, Restart, Proxy
Daemon username:	edguser
Process name(s):	funkydaemon
Process ID (PID) file(s):	/var/run/funkydaemon.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/log/funkydaemon.log
Ports:	none
Cron entries:	EDG_LOCATION/etc/cron/funky-cron (hourly)
Env. variables:	none
Other daemons contacted:	nonfunkydaemon
Link(s) to documentation:	http://www.edg.org/funky.pdf

See Table 8.2 for summary of installation information.

8.2.1 Post-LCFGng Manual Configuration

8.2.2 Testing and Debugging

8.3 MyProxy

Table 8.3: MyProxy Installation Information

Description:	Short description.
Supplier:	VDT
Bugzilla Category:	Bugzilla Name
Node type:	RB
License:	European DataGrid License
LCFG Components:	mylcfgconfig
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-wl-ns
SysV directives:	Start, Stop, Status, Restart, Proxy
Daemon username:	edguser
Process name(s):	funkydaemon
Process ID (PID) file(s):	/var/run/funkydaemon.pid
Lock file(s):	None
Log file(s):	EDG_LOCATION_VAR/log/funkydaemon.log
Ports:	none
Cron entries:	EDG_LOCATION/etc/cron/funky-cron (hourly)
Env. variables:	none
Other daemons contacted:	nonfunkydaemon
Link(s) to documentation:	http://www.edg.org/funky.pdf

See Table 8.3 for summary of installation information.

8.3.1 Post-LCFGng Manual Configuration

8.3.2 Testing and Debugging

8.4 MySQL

Table 8.4: MySQL Installation Information

Description:	Daemon for the MySQL database package.
Supplier:	MySQL AB
Bugzilla Category:	AAA: NONE OF THE OTHERS
Node type:	RLS (by WP2), MON and IC (by WP3)
License:	GPL (programs) / LGPL (old libraries)
LCFG Components:	None
Config. files:	None
Config. templates:	None
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	mysql
SysV directives:	start, stop, restart
Daemon username:	mysql
Process name(s):	mysqld_safe (1) mysqld (10s)
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	None by default
Ports:	3306
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://www.mysql.com/

See Table 8.4 for summary of installation information.

8.4.1 Post-LCFGng Manual Configuration

8.4.2 Testing and Debugging

To check for corruption of the MySQL databases, use the command:

```
mysqlcheck -u root -p --all-databases
```

There are additional flags which will also effect the necessary repairs. Occasionally queries hang. The command:

```
mysqladmin -u root -p processlist
```

allows you to check for such queries.

To activate the logging for MySQL. Create a file `/etc/my.cnf` which contains the following:

```
[mysqld]  
log = /var/lib/mysql/mysql.log
```



and also a file `/root/.my.cnf` which contains

```
[mysqladmin]
user = root
password = <password>
```

The first activates the logging and the second allows the `/etc/logrotate.d/mysql` to work.

8.5 Remote File IO (RFIO)

Table 8.5: RFIO Installation Information

Description:	RFIO daemon.
Supplier:	CASTOR (CERN)
Bugzilla Category:	RFIO
Node type:	SE
License:	GPL
LCFG Components:	none
Config. files:	/etc/shift.conf(?)
Config. templates:	n/a
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	/etc/init.d/rfiod
SysV directives:	start, stop, status, restart, condrestart
Daemon username:	root
Process name(s):	rfiod
Process ID (PID) file(s):	/var/run/rfiod.pid
Lock file(s):	/var/lock/subsys/rfiod
Log file(s):	/var/log/rfiod
Ports:	3147
Cron entries:	None
Env. variables:	none
Other daemons contacted:	None
Link(s) to documentation:	http://castor.web.cern.ch/castor/DOCUMENTATION/MAN/rfio/rfiod.man.html

See Table 8.5 for summary of installation information.

8.5.1 Post-LCFGng Manual Configuration

8.5.2 Testing and Debugging

8.6 Tomcat Daemon

Table 8.6: Tomcat Installation Information

Description:	Servlet Container
Supplier:	Apache
Bugzilla Category:	None
Node type:	IC, VOMS, RLS, SE, MON
License:	Apache License
LCFG Components:	edg-java-security
Config. files:	/etc/tomcat4/tomcat4.conf /etc/tomcat4/server.xml
Config. templates:	None
Needs CA certs./CRLs:	Yes
Needs host/service cert.:	Yes
Uses mapfile:	Yes
SysV script:	edg-tomcat4
SysV directives:	start, stop, restart, condrestart, proxy, proxy-destroy
Daemon username:	tomcat4
Process name(s):	java
Process ID (PID) file(s):	None
Lock file(s):	None
Log file(s):	/var/log/tomcat4/catalina.out
Ports:	8080, 8443, 9101–9120, 9201–9220, 9301–9320
Cron entries:	EDG_LOCATION/etc/cron/edg-java-security-container-proxy-renew-cron (12h)
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://jakarta.apache.org/tomcat/index.html http://edg-wp2.web.cern.ch/edg-wp2/security/trust-manager.htm

See Table 8.6 for summary of installation information.

8.6.1 Post-LCFGng Manual Configuration

8.6.2 Testing and Debugging

9 Testing Your Installation

A set of tests, included in the release, help verify the correct installation and configuration of your site. The edg-site-certification package contains these tests. All of the tests are invoked in a similar manner:

```
edg-testbed-test BaseTest::ReverseDNS alpha.example.org
```

where this would invoke the Reverse DNS lookup test for the host alpha.example.org. An overview of the tests can be found in the directory `/opt/edg/share/doc/edg-site-certification*`. To run these tests the directory `/opt/edg/bin` must be in your path and PERLLIB must include `/opt/edg/lib/perl`. Both should be done automatically if the EDG profile scripts are properly installed.

The EDG Users' Guide contains a set of simple examples which are also useful in testing your site. It also gives a useful overview of the testbed from the user's perspective.

Further information can be obtained from the Testing Group's web page¹.

¹<http://marianne.in2p3.fr/datagrid/TestPlan/>

10 Support

10.1 Contacts

The user's first point of contact for operational problems is the local site administrator. As such, you should try to answer questions from your local users and answer them if possible. System administrators and users are welcome and encouraged to use the bug-reporting facility. As a last resort, users may contact the Integration Team¹.

System administrators are encouraged to use routinely the bug-reporting system, Bugzilla², for flaws in the DataGrid software. For installation and configuration assistance, system administrators are first referred to the documentation but are welcome to contact the Integration Team³ directly.

10.2 Website

The main WP6 website⁴ contains documentation, contact information, the bug-reporting system, links to the source and packages repositories as well as links to other sites. This serves as the single point-of-access to information about the testbed activities.

¹<mailto:hep-proj-grid-integration-team@cern.ch>

²<http://marianne.in2p3.fr/datagrid/bugzilla/>

³<mailto:hep-proj-grid-integration-team@cern.ch>

⁴<http://marianne.in2p3.fr/>

11 Appendix

In these sections we collect information that might deepen the understanding or might help configuring certain components. In general this information is not needed to setup a EDG testbed.

11.1 Time Synchronization

Time plays a vital role when checking the validity of certificates. Consequently it is vital that the DataGrid machines be clients of a reliable time server.

If the machines at your site are not already synchronized, then you may use the `xntp3` package distributed with the other external packages used by EDG (See 3). This package implements the network time protocol (`ntp`)¹ and allows a machine to be a time client (as well as a time server).

If you use the `xntp3` package, then configuring your machine as a time client is rather trivial. You must add at least one time server reference to the `ntp` configuration file `/etc/ntp.conf` and configure the machine to run the `ntp` daemon. The detailed steps are:

- Identify one or more independent time servers to use. If your site does not have a standard time server, then consult the list of public time servers².
- The `ntp` protocol uses the port 123 (`tcp` and `udp`); ensure that your firewall will not filter these packets.
- Set the clock to the time of one of the servers

```
/usr/sbin/ntpdate ip-time-1.cern.ch
```

replacing the CERN time server with the one you have chosen.

- Synchronize the hardware clock on the machine to the system clock.

```
/sbin/hwclock --systohc
```

- For each time server, add an entry in the `/etc/ntp.conf` file like `"server ip-time-1.cern.ch"` replacing the CERN time server with the one you have chosen. You may specify multiple servers.
- Force the `ntp` daemon to start automatically at boot time.

```
/sbin/chkconfig xntpd on
```

- Start the `ntp` daemon

```
/sbin/service xntpd start
```

(or reboot the machine). You can check the status of the time server by using the `"/usr/sbin/ntpq"` command.

¹<http://www.eecis.udel.edu/ntp/>

²<http://www.eecis.udel.edu/mills/ntp/servers.htm>

Table 11.1: Spec File Changes

	architecture	Directory Name
Linux	linux	LINUX
Solaris 2.6, gcc	solaris-2.6-gcc	SUN4SOL2
SGI Irix 6.x, cc	irix-6-cc	SGI64

The `xntp3` package tries to be rather gentle to the system when readjusting the time. One result is that if the time is too far wrong (more than 1000s by default), then the daemon will simply refuse to reset the clock and will die. This is a common problem if you forget to use the `ntpdate` above.

It is extremely important that the hardware clock is synchronized to the system clock. If not, at the next boot an unsynchronized time will be reloaded and you risk having the time synchronization daemon stop.

If you have a large number of machines, you may wish to create a local time server. Refer to the local documentation of the `xntp3` package for instructions.

11.2 Monitoring

11.2.1 Network Monitoring

11.2.2 Application Monitoring with GRM/PROVE

For more information on GRM see GRM - Grid Application Monitor Users Manual. For more information on PROVE, see PROVE-Visualisation tool for Grid Applications.

For linux, installing the RPM does all necessary configuration.

For other operating systems, the following must be done. Replace the terms 'linux' and 'LINUX' in the "grm.spec" and "prove.spec" files with appropriate terms from 11.1. The 'linux' term signifies the architecture. 'LINUX' is the name of the (sub)directory that will contain the binary files for linux.

A configuration file with the same name ('linux.def') should be present in the `conf/` directory of the source. In the `conf/` file there are also configuration files for the irix and solaris operating systems as examples (`irix-6-cc.def` and `solaris-2.6-gcc.def`).

11.3 Installing Application Software

The software specific to various applications is available from the EDG package repository³. You should install all of the application software necessary to support the users authorized to use your site.

When installing application software be sure that you update the `RunTimeEnvironment` flags in `/opt/edg/info/mds/etc/ldif/ce-static.ldif/` and restart the information systems. This will publish via the information systems that you have installed the given set of software.

11.4 GSI FTP Daemon

A GSI-enabled daemon must run on any node which needs to serve its local file system to remote users via GridFTP (i.e. via the client `globus_url_copy` which uses `gridftp` as the transport protocol). This includes the gatekeeper, resource broker, and storage element nodes.

³<http://datagrid.in2p3.fr/pkgs/raw/applications/index.html>

Table 11.2: FTP Daemon Parameters

Parameter	Default	Description
GLOBUS_LOCATION	/opt/globus	Installation root of Globus software.
GLOBUS_GSIWUFTPD_PORT	2811	Port to use for GSI-enabled FTP.
GLOBUS_GSIWUFTPD_LOG	/var/log/globus-gsi_wuftpd.log	Location of log file.
X509_GSIWUFTPD_CERT	/etc/grid-security/hostcert.pem	Location of host certificate.
X509_GSIWUFTPD_KEY	/etc/grid-security/hostkey.pem	Location of host key.
GRID_GSIWUFTPD_USER	root	User to run FTP daemon.
GLOBUS_GSIWUFTPD_OPTIONS	unspecified	Additional FTP daemon options.
GLOBUS_TCP_PORT_RANGE	unspecified	Range of TCP ports (e.g. "30000,31000")
GLOBUS_UDP_PORT_RANGE	unspecified	Range of UDP ports (as above)

11.4.1 Security

Incoming requests are authorized via the grid-mapfile mechanism. Consequently, machines running the ftp daemon must have a full security installation. That is the machine must have a host certificate and key installed, a grid-mapfile, and all of the security RPMs which contain the Certificate Authority certificates and Certificate Revocation List URLs installed. The daemons which update the grid-mapfile (4.3.3) and CRLs (4.2.4) should also be running.

11.4.2 Configuration

The FTP daemon is configured via the `/etc/globus.conf` file. 11.2 lists the relevant parameters, their default values, and their descriptions.

11.4.3 Control

This daemon is controlled via a standard init.d-style script which support the start, stop, restart, and status directives.

12 Portable Batch System

12.1 Configuration

See Table 12.1 for summary of installation information.

Table 12.1: PBS Installation Information

Description:	Portable Batch System (PBS)
Supplier:	OpenPBS
Bugzilla Category:	None
Node type:	CE (server), WN (client)
License:	Open Source
LCFG Components:	pbsexehost pbsknownhosts
Config. files:	EDG_LOCATION/etc/??
Config. templates:	EDG_LOCATION/etc/??
Needs CA certs./CRLs:	No
Needs host/service cert.:	No
Uses mapfile:	No
SysV script:	pbs_server (server) pbs_sched (server) pbs_mom (server, client)
SysV directives:	start, stop, status
Daemon username:	root
Process name(s):	??
Process ID (PID) file(s):	??
Lock file(s):	??
Log file(s):	??
Ports:	none
Cron entries:	None
Env. variables:	None
Other daemons contacted:	None
Link(s) to documentation:	http://www.openpbs.org/

See this script¹ for an example implementation of the following description

Much of the configuration for the PBS server and all of the configuration for the PBS clients is handled by the LCFG components. Nonetheless this section describes those steps necessary to get PBS running on your cluster. For detailed configuration instructions for OpenPBS consult the manual.

EDG redistributes OpenPBS rpms from the package repository. The current version in use is listed in the EDG release rpm lists. These rpms have many of the outstanding PBS patches applied and they also install the PBS executables in more typical locations in the linux file system.

There are three PBS daemons—`pbs_server`, `pbs_sched`, and `pbs_mom`. The first two are installed on the PBS master node (the gatekeeper node for EDG) and the last on the PBS clients. The gatekeeper can be treated as a PBS client as well, if you wish. Use **chkconfig** to enable these daemons at startup:

¹http://gridportal.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp4/sites/fzk/other/tb_postinst_scripts/pbs-server



```
/sbin/chkconfig --level 345 pbs_server on
/sbin/chkconfig --level 345 pbs_sched on
/sbin/chkconfig --level 345 pbs_mom on
```

and verify this is the case with:

```
/sbin/chkconfig --list
```

If you're using LCFG this will have already been done.

Before the configuration, turn off these daemons:

```
/sbin/service pbs_server stop
/sbin/service pbs_sched stop
/sbin/service pbs_mom stop
```

If the file `/var/spool/pbs/pbs_environment` does not exist then create it with the following command:

```
touch /var/spool/pbs/pbs\_environment
/usr/sbin/pbs\_server -t create
```

You will have to stop the PBS server daemon again. Edit this file so that it contains the following:

```
PATH=/bin:/usr/bin:/usr/local/bin:/usr/sbin
LANG=en_US
LC_ALL=en_US
```

This defines the environment used by the PBS jobs.

Next create the file `/var/spool/pbs/server_priv/nodes`. This file contains a list of your PBS clients. There is a one line entry per machine:

```
pbsclient1.example.org np=2 laldev
```

The “np=2” defines the number of virtual processors on the machine and the “laldev” is an arbitrary name for your cluster; you should use the fully-qualified host name. You should *not* mark the machines as time-shared (“:ts”) nodes.

Put the fully-qualified host name of the server in the file `/var/spool/pbs/server_name` and start the PBS server and scheduler:

```
/sbin/service pbs_server start
/sbin/service pbs_sched start
```

Now create a file, `pbs.conf`, with the following contents, customizing the email addresses as necessary.

```
#
# Create and define queue short
#
create queue short
set queue short queue_type = Execution
set queue short resources_max.cput = 00:15:00
set queue short enabled = True
set queue short started = True
#
# Set server attributes.
```



```
#
set server scheduling = True
set server managers = root@grid01.lal.in2p3.fr
set server operators = root@grid01.lal.in2p3.fr
set server default_queue = short
set server log_events = 511
set server mail_from = adm
set server query_other_jobs = True
set server scheduler_iteration = 100
set server node_pack = False
```

Feed this to the PBS server with the following command:

```
qmgr < pbs.conf
```

This will create one batch queue named “short” with the given parameters and activate the server.

The PBS client configuration is fully handled by the LCFG configuration object **pbsexechost**. However for a manual installation create the file `/var/spool/pbs/mom_priv/config` with the following contents:

```
$clienthost localhost
$clienthost pbsclient1.example.org
$restricted pbsserv.example.org
$logevent 255
$ideal_load 1.6
$max_load 2.1
$usecp *.example.org:/home /home
```

The last line is important as it instructs PBS to use a **cp** command to return the job’s standard output and error to the server rather than a remote copy command **rcp**. For the grid software to work, there must be a shared file system between the PBS server and PBS clients. With

```
/sbin/service pbs_mom start
```

start the PBS client server.

12.2 Testing

To test that the PBS configuration is working, log onto the PBS server as a regular user. Create a test script `pbstest.sh` like the following:

```
#!/bin/sh
hostname
date
sleep 5
date
```

Submitting this with the **qsub pbstest.sh** command should return a batch identifier. You can check that this is running with the **qstat -Q** command. When it finishes, you should have two files `pbstest.sh.e0` and `pbstest.sh.o0`. (Trailing number may be different.) The error file (`*.e0`) should be empty and the standard output (`*.o0`) should have the output from the script.



If you cannot submit the job, most likely some part of the above configuration was not followed correctly or one of the daemons is not running. If the job does not return the standard error and output, then likely there is some problem with the shared file system between the server and clients.

Globus places one additional constraint on the system: the user accounts between server and client must be shared and must have a shared ssh key. These keys are generated via the **poolaccounts** configuration object for LCFG clients. They must be made by hand in a manual configuration. The known hosts file must also contain the keys of all of the clients and the server; this is handled via the command `edg-pbs-knownhosts` located in the `/opt/edg/sbin` directory. To test if the configuration works correctly do the following:

```
ssh pbsclient1.example.org /bin/echo OK
```

as a non-privileged user. This should return the string "OK" without any interactive prompts. If you are prompted to accept the host key, the configuration is not correct. If you forget to do this, grid jobs will fail with the error "Host key validation failed" even while PBS is functioning correctly.