Guide to Job Repository version 1.0.4

Oscar Koeroo

5 March 2004

1 Introduction

The Gridification subtask of WP4 of the European Datagrid project[1] interfaces the local fabric to other middleware components by a number of services, among which the Local Centre Authorization Service (LCAS) handles authorization requests to the local computing fabric and the Local Credential Mapping Service (LCMAPS) provides all local credentials needed for jobs allowed into the fabric. The Job Repository collects the user, job and user-mapping information when a job has been assigned to a fabric and handled by LCMAPS. This document describes the Job Repository, which is an extra functionality to log and keep track of a fabric, released by the Gridification subtask.

The Job Repository consists of different parts:

- `MySQL database`, a database that holds the information as the Job Repository database which is connectable via (My)ODBC.
- `Job Repository LCMAPS module`, this module is loaded in the LCMAPS framework and extracts job, user and credential information and stores this information in the Job Repository database.
- `Job Repository API`, this library is the interface between the LCMAPS module and the ODBC connection to the database.
- `Job Repository Utils`, the utils contain scripts to execute database creation, recreation and dropping. It also contains script (written in Perl with the DBI lib.) that can update a Job status (to be executed by a Job Manager) and a script that can translate a FQAN in to a Unix Group ID according to it's registered mapping.

[1] http://www.eu-datagrid.org

More information on the LCMAPS and other components of the Gridification subsystem can be found in:

- the WP4 architecture document D4.2: pdf version[2] or doc version[3].
- the description of the Job Repository API: here[4] , PostScript file[5] and PDF file[6].
- LCMAPS: http://www.dutchgrid.nl/DataGrid/wp4/lcmaps/[7]
- LCAS: http://www.dutchgrid.nl/DataGrid/wp4/lcas/[8]
- the README[9], INSTALL[10], and LICENSE[11] files.

2 Installation

As stated in the introductory we will split the installation in two subsections. The first section will handle the Database side. The other will handle the CE side of the installation.

2.1 The Database

The Job Repository needs to have a SQL database server. We build it around a MySQL database so this document will only state this database. Also the database creation scripts are written in a (possibly) MySQL specific style. Take this into account if you wish to upgrade and/or alter the database server into something else like Postgres or Oracle..

The packages listed here are needed for the backend side of the Job Repository:

The Job Repository comes with a 'Utils' (edg-jobrepository_gcc3_2_2-utils) package. This package contains several scripts. Three of them are

[2] http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/architecture/eu/WP4-architecture-2_1.pdf

[3] http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/architecture/eu/WP4-architecture-2_1.doc

[4] apidoc/html/index.html

[5] apidoc/latex/refman.ps

[6] apidoc/latex/refman.pdf

[7] http://www.dutchgrid.nl/DataGrid/wp4/lcmaps

[8] http://www.dutchgrid.nl/DataGrid/wp4/lcas

[9] README

[10] INSTALL

[11] LICENSE

Table 1: RPMs to be installed on the database running machine.

RPM	min. version	description + URL
MySQL-server	4.0.13	the MySQL backend for the job repository http://datagrid.in2p3.fr/distribution/external/RPMS/
MySQL-client	4.0.13	the MySQL client, needed by the job repository utils http://datagrid.in2p3.fr/distribution/external/RPMS/
MySQL-shared-compat	4.0.13	the compatibility lib, needed for MyODBC http://datagrid.in2p3.fr/distribution/external/RPMS/
MyODBC	3.51.06	the MySQL ODBC 3.51 Driver http://datagrid.in2p3.fr/distribution/external/RPMS/
edg-jobrepository_gcc3_2_2-utils	1.0.4	This package contains utility scripts to obtain relevant info from the job repository http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RPMS/

interessting to be used on the machine that will have the MySQL daemon or has the MySQL client software installed and the abbility to connect to a server from commandline. The scripts of interest

- `edg-jobrep-database-create`, Creation script of the database and it's tables and user with the right priviledges.
- `edg-jobrep-database-drop-all`, Ereases the database. All data will be lost. This has only testing and maintanance purposes.
- `edg-jobrep-database-recreate`, Executes the drop script, followed by the creation script. The same warning as with the '*drop-all' script is at hand. All data will be lost.

These scripts will execute the SQL creation or drop scripts with help of the MySQL commandline interface to the database. Therefor you'll need a MySQL client to execute the scripts.

The Job Repository LCMAPS module

The Job Repository can not exist without the parts that fill the database.

The packages listed here are needed for the frontend side of the Job Repository:

Note: The MyODBC has to be installed clientside because of a database driver that comes with the package to make a successfull connection to the ODBC coupling on the backend. The iODBC packages provides an '/etc/odbc.ini' file or a '.odbc.ini' in root's home-directory. You can use them by filling in the needed configuration options with a DSN name for this database connection. This prevents you from having the need to set these configuration parameters in other configuration files limiting the configuration parameters in the other files to just the DSN name of the connection. This is optional. We recommend that all the configuration is done in the 'lcmaps.db' file located by default at '/opt/edg/etc/lcmaps/lcmaps.db'. This

Table 2: RPMs to be installed on the CE machine.

RPM	min. version	description + URL
edg_gatekeeper_gcc3_2_2-gcc32dbg_pgm	2.2.14	the modified globus gatekeeper http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RP
edg-lcmaps	0.0.23	the LCMAPS library (= pluginframework + utilities) and an example LCMAPS plugin http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RP
edg-lcmaps-basic_plugins	0.0.23	the LCMAPS plugins providing the basic globus-gatekeeper functionality http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RP
edg-lcmaps-voms_plugins	0.0.23	the LCMAPS plugins that base the credential mapping on the VO information inside the http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RP
edg-lcmaps-jobrep_plugins	0.0.23	the LCMAPS plugin that stores the lcmaps info in the job repository http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RP
libiodbc	3.51.1-1.edg1	the iODBC Driver Manager http://datagrid.in2p3.fr/distribution/external/RPMS/
MyODBC	3.51.06	the MySQL ODBC 3.51 Driver http://datagrid.in2p3.fr/distribution/external/RPMS/
edg-jobrepository_gcc3_2_2-api	1.0.4	This package contains the api of the job repository http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3-gcc3.2.2/wp4/gridification/RP

'lcmaps.db' file contains the configuration and localsite policy of the CE for the LCMAPS system. Keeping the Job Repository initialisation string in the lcmaps.db will give you a more centralized sense on the configuration. There is no need for the odbc.ini files to be used. Also the odbc.ini files gives a security valnurability for the database is the odbc.ini file isn't in a root-only environment. Using passwords in the lcmaps.db file gives the system a security comprimise for the Username and Password that will be used by the LCMAPS module of the Job Repository. Although still allowed it is needless to say that this must be done differently. The functionally to insert a Username and Password in the lcmaps.db only has debugging and testing purposes.

A file called 'jobrep_config' with root-read-only (0400) priviledges is strongly adviced to make in '/opt/edg/etc/lcmaps/'. In the initialisation string of the LCMAPS Job Repository Plugin you can specify with "-JR_CONFIG ¡path/file¿" where this file is located. The file must have a 0400 root-read-only file description.

`The creation of this 'jobrep_config file in (default path:)  '/opt/edg/etc is a Manual action`;

3 Configuration

3.1 LCMAPS

LCMAPS is configured generally in the lcmaps.db file (by default located at '/opt/edg/etc/lcmaps/'. As any other module it must specified in the module specification part of the LCMAPS policy file. For a more detailed

explenation about using LCMAPS and it's policy file and the configuration of it you can take a look at: LCMAPS[12]

There are a few rules to follow when using the LCMAPS Job Repository plugin. Let's start at the end. In the policy file you'll notice to sections. The first is for setting the path, the module's (a shared object) filename and the initialisation string. In the second section you'll see the policies defined. Here is an example of the policy section with one policy:

```
#My example1
example:
vomsextract -> vomslocalgroup
vomslocalgroup -> vomspoolaccount
vomspoolaccount -> posixenf
```

Now I'll show you the same example only now it is extended with the Job Repository module:

```
#My example2
example:
vomsextract -> vomslocalgroup
vomslocalgroup -> vomspoolaccount
vomspoolaccount -> jobrep
jobrep -> posixenf
```

As you can see, the 'jobrep' is to correspond with the Job Repository module and it's initialisation string. The Job Repository will get all the information available on that point in time when it is called in the policy evaluation procedure. The Job Repository plugin will get information that has been gathered and stored into the framework that other plugins gathered for it. In this example I used a policy that will map a user according to his/hers VO(s). The Job Repository plugin will also work fine without the VO information in a user's certificate. Here is a example for policies that uses the basic mapping methods:

```
#My example3
example:
localaccount -> jobrep | poolaccount
poolaccount -> jobrep
jobrep -> posixenf
```

[12] ../lcmaps/index.html

Again the 'jobrep' will be called before the actual enforcement. When you have a site that want a more dynamic handling of the users in a fabric then LDAP with the VOMSPoolGroups module comes in handy. A policy for that would change into the following when you use 'jobrep':

```
#My example4
example:
vomsextract -> vomspoolgroup
vomspoolgroup -> vomspoolaccount
vomspoolaccount -> jobrep
jobrep -> ldapenf
ldapenf -> posixenf
```

To work back up in the policy file we will see the initialization string of the LCMAPS plugins, including the one for the Job Repository. Here are the initialisation options for the 'jobrep' module (note: jobrep is an alias for the module and it's parameter, this could be anything):

- `lcmaps_jobrep.mod`: this plugin ... More info ...

Here is an example of a nice working initialisation string on our development testbed: `jobrep = "lcmaps_jobrep.mod -vomsdir /etc/grid-security/vomsdir/ -certdir /etc/grid-security/certificates/ -jr_config /opt/edg/etc/lcmaps/jobr server=tbn.nikhef.nl driver=/usr/local/lib/libmyodbc3.so database=JobReposito`

3.2 The Database

We assume that the needed packages for the backend host are installed and working properly. The `edg-jobrepository_gcc3_2_2-utils` RPM provides `/opt/edg/sbin/edg-jobrep-database-create`, `/opt/edg/sbin/edg-jobrep-database-dr` and `/opt/edg/sbin/edg-jobrep-database-recreate` for creating, dropping and recreating a Job Repository database. The scripts will call the mysql client software and will execute the accomponing '*.sql' scripts to make or erase a working default installation. Note: dropping a database will remove all database tables from the server and the database itself. Take care with this functionality because it will result into a clean and empty (unrecoverable) database. By default there are grants made for two additional users in the database. The first is meant to be used by the LCMAPS module. It's password (and username) must be unreadable to the outside world. It may only `Insert`, `Select` and `Update` information in the database. This user may only execute the gaven privileges within the domain (like

To configure the database you must execute `edg-jobrep-database-create`. The `edg-jobrepository_gcc3_2_2-utils` RPM installs a Man page here /opt/edg/share/man/man1/edg-jobrep-database-create.1 that will guide you to make a Job Repository database.

This has to be done `manually` because it contains the setting of the username and password for the LCMAPS module and you will have to fill in the password for the root database-user.

LCFG configuration:

In the Computing Element resource file ComputingElement-cfg.h[13] has to be update with a new LCMAPS policy. The LCMAPS policy (default path: /opt/edg/etc/lcmaps/lcmaps.db) has to be updated with the inclusion of the LCMAPS module in the policy execution. Note: Policy fall-over will accure if one policy fails and an other one is defined. So is the Job Repository would fail or could be so that another policy still mapped a user to the site.

Further more, the creation of the 'jobrep_config file in (default path:) '/opt/edg/etc/lcmaps/' is a `manual` action. This is where the database password (and/or username) must be available (and correcly configured in the LCMAPS policy for the initialization of the Job Rep. module) to autorise itself into the MySQL database.

Last but not least (actually it will probably be more handy to execute this first...) the 'edg-jobrep-database-create' script (default location: /opt/edg/sbin/edg-jobrep-database-create) must be run on the database running machine. This will create the database within the MySQL server.

[13] http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/edg-release/ng_source/ComputingElement-cfg.h