Guide to Job Repository version 1.3.3

Oscar Koeroo

21 October 2005

1 Introduction

This component is constructed as an effort of the Joint Research Activity 3 (JRA3) which is responsable for all security aspects of the Enabling Grids for E-sciencE (EGEE)[1]. This component is part of the security subsystem of the gLite software distrobution.

The Job Repository is orginally constructed as a part of the European Datagrid project[2] project. It was constructed as a part of the Gridification subtask (Work Package 4).

It collects information about the user, its 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 what is happening inside your farm.

The Job Repository consists of different parts:

- `The Database`, typicaly a MySQL database that holds the information as the Job Repository database which is connectable via an ODBC interface to ensure the interoperability with other then MySQL database backends.
- `Job Repository LCMAPS module`, this module is loaded in the LCMAPS framework and extracts job, user and credential information from the LCMAPS framework and stores this information in a relational database. It includes the database schema.
- `Job Repository API`, this library is the interface between the LCMAPS module and the ODBC connection to the database.

More information on LCMAPS can be found here[3].

[1] http://www.glite.org
[2] http://www.eu-datagrid.org
[3] http://www.nikhef.nl/grid/lcaslcmaps/

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

RPM	min. version	description + URL	
MySQL-server	4.0.13 (or higher)	the MySQL backend for the job repository	http://www.mysql.com
MySQL-client	4.0.13 (or higher)	the MySQL client, needed by the job repository utils	http://www.mysql.com
MyODBC	3.51.06 (or higher)	the MySQL ODBC 3.51 Driver	http://www.mysql.com

2 Installation

The installation guide is split in two subsections. The first will handle the details about setting up a database. It will describe how to handle with a MySQL database. Oracle and other databases will work simularly when using a different ODBC driver for the connection. The second subsection will explain the changes to be done to let the Job Repository work on your service. This concerns changing LCMAPS configurations and adding new configuration files to your system. At the present time only a CE is supported as service through the use of the LCMAPS framework. Other more common interfaces will not be realised within the EGEE project.

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 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 LCMAPS plugin rpm comes with four scripts. Only the 'jobrep-admin' script needs execution. The others are seperate SQL scripts to create the database. They are located in the (standard) location '/opt/glite/sbin/'. The scripts are:

- `jobrep-admin`, Script to install or remove a Job Repository database through the provided SQL scripts.

- `jobrep-create-basic.sql`, SQL script to install the default Job Repository DB schema. The schema supports more than one type of service to hold relational information from.

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

RPM	min. version	description + URL	
lcmaps	1.2.x (or higher (should also work fine with very old LCMAPS version))	the LCMAPS library (= pluginframework + utilities)	¡LCMAPS link¿
lcmaps-plugins-jobrep_plugins	1.3.x (or higher)	the LCMAPS plugin that stores the lcmaps info in the job repository	¡LCMAPS jobrep plugin¿
unixODBC	unixODBC-2.2.8	the unixODBC libraries	http://www.unixodbc.org
MyODBC	3.51.06 (or higher)	the MySQL ODBC 3.51 Driver	http://www.mysql.com
jobrepository	1.2.x (or higher)	This package contains the api of the job repository	¡jobrep api¿

- `jobrep-create-CE.sql`, SQL script to install the Compute Element specific tables needed to store relevant relational information about the CE activities.
- `jobrep-drop-total-database.sql`, SQL script to remove a Job Repository database completely.

2.2 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 both on the clientside and on the serverside. The clientside needs a database driver that comes with the package to make a connection to a MySQL database on the backend.

When all these rpms are installed then we can move to the (re)configuration of this system.

3 Configuration

3.1 LCMAPS

LCMAPS is configured generally in the lcmaps.db file (by default located at '/opt/glite/etc/lcmaps/'. This file is used by the LCMAPS framework to specify:

- The physical location on the disk of each modules (aka the plugins)
- Which initialization string must be used for each module
- The policies that are composed modules executed as a state machine. The outcome of each module (true or false) will decide which module

to execute next. This means they can either return true or false which will create a controlflow through the policies.

For a more detailed explenation about using LCMAPS and it's policy file and the configuration of it you can take a look at the LCMAPS[4] specific information pages.

To start at the top of the policy file we will see the initialization string of the LCMAPS modules, 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 you like) :

Here is an example of a working initialisation string:

```
jobrep = "lcmaps_jobrep.mod"
         "-jr\_config /opt/glite/etc/lcmaps/jobrep\_config"
```

3.2 The policies

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:
vomslocalgroup -> vomspoolaccount
vomspoolaccount -> jobrep
jobrep -> posixenf
```

As you can see, the 'jobrep' is to correspond with the Job Repository module and its 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 the enforcement fase. In this example I used a policy that will map a

[4]http://www.nikhef.nl/grid/lcaslcmaps/

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
```

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:
vomspoolgroup -> vomspoolaccount
vomspoolaccount -> jobrep
jobrep -> ldapenf
ldapenf -> posixenf
```

3.3 The 'jobrep_config' file

A file called 'jobrep_config' with (usually) root.root read-only (0400) privileges. In the initialisation string of the LCMAPS Job Repository Plugin you ca n specify with "-jobrep_config ¡path¿" where this file is located. The file must have these strict perimissions because this is the safest place to store t he database, username and password combination. The system administrator has the freedom not to use the file as such, but we urge you to do so because every authorized Grid user can (in most cases) access the directory '/opt/glite/etc/lcmaps/'. The database can hold non-disclosure information about the users an d there jobs that are running and that have run on your farm.

All the needed 'Database connection parameters' can be located in the '-jr_config' file which are:

- `username=`¡db username¿
- `password=`¡db password¿
- `database=`¡db database name¿

More detailed and needed configuration options will be shown in the ODBC specific odbc.ini file.

3.4 The odbc.ini file(s)

An ODBC initialization file will configure Data Source Names (DSNs) and link this name with a set of configuration parameters. Here is an example of such a file:

```
[ODBC Data Sources]
jobrepository = mysql ODBC Driver

[jobrepository]
Driver=/usr/lib/libmyodbc3.so
Port=3306
```

The 'ODBC Data Sources' is an reserved section title where all the DSNs are specified. As shown here the 'jobrepository' is the only DSN and has the identifiable specification to be an 'mysql ODBC Driver'. Below that line my jobrepository's set of parameters is given. In this example the set of values is at its minimum. Actually the port number used to connect to the server (localhost) is the default port and can be thrown out too. The share-object also known as the Driver needs to be specified. This is needed for the ODBC coupling to work and talk the correct dialect of database talk to the MySQL daemon.

Allowed other parameters are:

- `server=`¡host with the database other then localhost¿
- `username=`¡db username¿
- `password=`¡db password¿
- `database=`¡db name used for the Job Repository¿
- and others which could even be specific per database type and version

The ODBC initialization information can be stored in the default locations '/etc/odbc.ini' or in the private space of the user (or admin) ' /.odbc.ini'.

3.5 The Database

The Job Repository LCMAPS plugin rpm comes with four scripts. Only the 'jobrep-admin' script needs execution. The others are seperate SQL scripts to create the database. They are located in the (standard) location '/opt/glite/sbin/'. The scripts* are:

- `jobrep-admin`, Script to install or remove a Job Repository database through the provided SQL scripts.
- `jobrep-create-basic.sql`, SQL script to install the default Job Repository DB schema. The schema supports more than one type of service to hold relational information from.
- `jobrep-create-CE.sql`, SQL script to install the Compute Element specific tables needed to store relevant relational information about the CE activities.
- `jobrep-drop-total-database.sql`, SQL script to remove a Job Repository database completely.

* = The database schema is database specific. At first only MySQL will be supported. Other backends can be produced.

The 'jobrep-admin' script will guide you to setup the schema in a new database or to remove it dropping the database. Do think twice before you click the removal will result into wiped database unrecoverable by common tools.

By default there is a grant made one additional users in the database. That account is used by the LCMAPS plugin to work with the 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. The username and password for the LCMAPS module must be in sync with that of the 'jr_config' file.