

# Browsing Grid resources

## Browsing Grid Resources

It is possible for users and administrators of a Grid system to find out information about various resources of a Grid. The Information Service (IS) provides information about the EGEE/LCG-2 Grid resources and their status [3.1]2. It is through the information service that available CEs2 to run jobs can be located, and the SEs2 holding replicas of Grid files and the catalogs keeping the information on these files are found. This published information is also important for monitoring and accounting purposes.

## The GLUE Schema

The data published by IS conforms to the GLUE (Grid Laboratory for a Uniform Environment) schema [3.2]2. The GLUE schema defines a common conceptual data model to be used for Grid resource monitoring and discovery. There are three main components of the GLUE schema; they describe the attributes and values of CEs2, SEs2 and binding information for CEs2 and SEs2.

EGEE currently uses the Monitoring and Discovery Service (MDS) by Globus as the main provider of the information service. The new Relational Grid Monitoring Architecture (R-GMA) has also been deployed in EGEE/LCG-2 type Grids, however our focus will only be on MDS.

## Monitoring and Discovery Service

The MDS implements the GLUE Schema using OpenLDAP2; an Open Source implementation of the Lightweight Directory Access Protocol (LDAP) [3.3, 3.4, 3.5]2. LDAP is a special form of a database which is optimised for reading, browsing and searching data. LDAP information model is based on entries and those entries describe objects such as employees of a company, computers, and servers. Each of these entries (i.e. objects) contains one or more attributes that describes the entry. Each of these attributes has a type and one or more values. Each entry in the LDAP is uniquely identified by a Distinguish Name (DN). A DN is formed by sequence of attributes and values. Based on the DNs2 the objects can be arranged into a hierarchical tree like structure called the Directory Information Tree (DIT). Figure 3.1 shows an example of a hypothetical DIT.

Figure 3.1 – An example of a hypothetical directory information tree.

In figure 3.1 the root entry defines the site and site specific information is given as a separate entry in the next level. Similarly, CEs2 and SEs2 under the site is in the second level. Then SE specific entries such as file catalog and disk status is in the third level. The DN of a particular CE is formed by attribute identifying the site (Site\_ID=gridats) and attribute identifying the CE (CE\_ID=baciucogrid.it). Therefore the complete DN would be similar to CE\_ID=baciucogrid.it,Site\_ID=gridats.

Figure 3.2 – DIT of the Storage Resources

An actual DIT of a Grid is much more complex than figure 3.1. Just to give you a brief idea, let us look at a more realistic DIT that depicts SEs2 of a Grid. Figure 3.2 shows the GLUE schema DIT for storage resources at a site, as published by the site GRIS. All of the site entries are grouped under the "o=grid,mds-vo-name=local" root entries. For each SE host at the site, a "GlueSEUniqueID2" entry is present, and the value of the "GlueSEUniqueID2" is exactly the SE hostname; under each of these "GlueSEUniqueID2" entries are one or more "GlueSARoot2" entries, specifying the path(s), in the SE host file system, where Grid data is stored. The full DIT is much more complicated and comprises entries for the computing resources, and other services (LFC, RB, etc.).

The kind of information can be stored as entries in the DIT are specified in the LDAP Schema. The schema defines object classes, their attributes and value types. The EGEE implementation of OpenLDAP does not require any authentication (i.e. all users can browse the catalog and all services are allowed to enter or modify information into it).

Figure 3.3 – The MDS information service

Figure 3.3 show how the information is stored and propagated in a EGEE-based Grid. On the CE and SEs runs a program named "Information Provider", which monitors local resources and publishes entries describing their status (both static and dynamic) on the local GRIS LDAP database. The Grid Resource Information Server (GRIS) publishes this status information via an LDAP server.

In each site an element called the Berkeley Database Information Index (BDII) [3.6] compiles all the data from different GRISs and publishes it (from user perspective, there is no difference between a BDII and GRIS). The current implementation of EGEE recommends the use of Berkeley Database Information Index (BDII) [3.6] due to its higher stability. This BDII is called the Site BDII and make use of LDAP.

A BDII is also used at the top of the IS hierarchy. This BDII queries all the GIISes (or site BDIIs) and act as a cache storing information about the Grid status in its database. Therefore by querying the top most BDII a user or a service can access all the available information about the Grid. It is also possible to get information about a specific resource by querying individual site GIISes, site BDIIs or local GRISs. This information can be up-to-date compared to what is in the top BDII.

## The lcg-infosites command

The lcg-infosites command can be used as an easy way to retrieve information on grid resources for the most common user needs. The command has the following format; meaning of attributes and options are summarised in table 3.1:

```
lcg-infosites --vo [vo name] request -v [verbose level] -h --is [DBII to query]
```

To get a detailed description on lcg-infosites command use the following:

```
$ lcg-infosites -h
```

Table 3.1 – Supported requests of the lcg-infosites command

Request	Description
all	It groups together the information provided by CE, SE, LRC and RMC.
ce	The information relative to number of CPUs, running jobs, waiting jobs and names of the CEs are provided.
closeSE	The names of the CEs where the user's VO is allowed to run together with their corresponding closest SEs are provided.
lfc	The name of the machine hosting the LFC catalog is printed.
lrc rmc	<b><i>The LRC/RMC services are obsolete.</i></b>
se	The names of the SEs supported by the user's VO together with their types, available & used space are printed.
tag	The names of the tags relative to the software installed in site are printed together with the corresponding CEs.

Table 3.2 – Options of lcg-infosites command

Option	Description
<code>--vo</code>	Name of the VO (mandatory)
<code>-v</code>	<p>For Computing Elements:</p> <p><code>-v 1</code> Print only the names of the queues.</p> <p><code>-v 2</code> Print also the names of the queues together with the RAM Memory, Operating System and its version and the processors are printed.</p> <p>For Storage Elements:</p> <p><code>-v 1</code> Print only the names of the SEs?.</p>
<code>-h, --help</code>	Display help
<code>--is</code>	BDII user wishes to query. In case this argument is not supplied, the BDII defined into the LCG_GFAL_INFOSYS environment variable will be queried.

## Listing CEs? and queues

For instance, let us try to obtain information about computing resources of `gridats`. Use the following command:

```
$ lcg-infosites --vo gridats ce
```

A typical output is as follows:

```
*****
These are the related data for gridats: (in terms of queues and CPUs)
*****

#CPU Free Total Jobs Running Waiting ComputingElement
-----
 38  38   0         0         0  grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-grid
  4   2   0         2         0  ce-01.grid.sissa.it:2119/jobmanager-lcgpbs-gridats
 10   2   0         0         0  baciuco.grid.sissa.it:2119/jobmanager-pbs-gridats
 10  10   0         0         0  egrid.create-net.org:2119/jobmanager-lcgpbs-gridats
  8   0   0         0         0  egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats
```

The verbose level option `-v` can be used to control the amount of information displayed on screen. The verbose level is given by either an integer. The higher the number, the more amount of information is displayed.

Compare the following outputs of `lcg-infosites ce` at verbose levels 1 and 2.

```
$ lcg-infosites --vo gridats -v 1 ce
ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-long
grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-gridats
ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-short
ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-infinite
egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats
gridts01.grid.elettra.trieste.it:2119/jobmanager-lcgpbs-gridats
```

```
$ lcg-infosites --vo gridats -v 2 ce
RAMMemory      Operating System  System Version      Processor      Subcluster name
```

---

513	Scientific Linux SL	3.0.7	PIII
2048	Scientific Linux CERN	SLC	PIII
513	Scientific Linux SL	3.0.7	PIII
1025228	SL	3	PIII
1024	SLC	3	PIV

## Listing SEs?

Retrieve the information about storage elements using the following command:

```
$ lcg-infosites --vo gridats se
```

A typical output is as follows:

```
Avail Space(Kb) Used Space(Kb) Type SEs
-----
688004000      874446272      n.a  grid002.oat.ts.astro.it
13659804       1054456         n.a  egrid-ce-01.egrid.it
169452728      7753952         n.a  gridts01.grid.elettra.trieste.it
```

The verbose level option `-v` can be used to control the amount of information displayed on screen. The verbose level is given by either an integer. The higher the number, the more amount of information is displayed.

The SE listing only admits verbose level 1:

```
$ lcg-infosites --vo gridats -v 1 se
grid002.oat.ts.astro.it
egrid-ce-01.egrid.it
gridts01.grid.elettra.trieste.it
```

## Listing CEs? and SEs? together

The request `all` can be used to retrieve information on both CEs? and SEs? at the same time. If you type:

```
$ lcg-infosites --vo gridats all
```

Then you will see results for the two sections above appearing one after another.

## Inspecting the CEs? and SEs? proximity relation

With the request `closeSE`, you can retrieve the names of the CEs? where our VO (in this case `gridats`) is allowed to run together with the corresponding closest SEs?. Use the following command:

```
$ lcg-infosites --vo gridats closeSE
```

A typical output is as follows:

```
Name of the CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-long
                ce-02.grid.sissa.it

Name of the CE: grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-gridats
                grid002.oat.ts.astro.it

Name of the CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-short
```

```
ce-02.grid.sissa.it
```

```
Name of the CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-infinite  
ce-02.grid.sissa.it
```

```
Name of the CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats  
egrid-ce-01.egrid.it
```

```
Name of the CE: gridts01.grid.elettra.trieste.it:2119/jobmanager-lcgpbs-gridats  
gridts01.grid.elettra.trieste.it
```

## Listing the LFC server hostname

The name of the LFC (LCG File Catalog) from a given Grid (i.e. `gridats`) can be found out using the `lfc` request. Use the following command:

```
$ lcg-infosites --vo gridats lfc  
gridts05.grid.elettra.trieste.it
```

## Listing installed software tags

The name of the tags relative to the software installed in a site (i.e. `gridats`) together with the corresponding CEs<sup>2</sup> can be printed using the tag option. Following is an illustration of the command (we use the `egrid` VO, since `gridats` uses no software tags at present):

```
$ lcg-infosites --vo egrid tag  
valor del bdii: egee-bdii.cnaf.infn.it:2170  
Name of the CE: gridce.ilc.cnr.it
```

```
Name of the CE: gridba2.ba.infn.it  
VO-egrid-client
```

```
Name of the CE: grid002.ca.infn.it
```

```
Name of the CE: grid012.ct.infn.it  
VO-egrid-Python_2.3_1  
VO-egrid-Python-2.3  
VO-egrid-uberftp-VDT1.2.0  
VO-egrid-eurl  
VO-egrid-egrid-transform-dispatcher  
VO-egrid-transform-nyse-taq  
VO-egrid-etool  
VO-egrid-scilab-4.0  
VO-egrid-octave-2.1.73  
VO-egrid-scilab  
VO-egrid-octave  
VO-egrid-client
```

```
[.....]
```

## The `lcg-info` command

The `lcg-info` command can be used to list either CEs<sup>2</sup> or SEs<sup>2</sup> that satisfy a given set of conditions on their attributes, and to print, for each of them the values of a given set of attributes. The information is taken from the BDII specified by the `LCG_GFAL_INFOSYS` environment variable or given through the command line. The command has the following format and meaning of each option is summarised in table 3.3:

```
lcg-info [--list-ce] [--list-se] [--query
```

```
<user query>] [--attrs <user attributes>] -vo
<vo name> --sed --help --list-attrs
```

Note

*Note:*

If `--list-ce` is specified then only CE attributes are considered and if only `--list-se` is specified then only SE attributes are considered, not both. The attributes used with `--attrs` or used within `--query` expressions are a subset of GLUE schema, those that are most relevant to the user.

Note

*Note:*

To use the `lcg-info` command Perl-based LDAP packages (for the `Net::LDAP PERL` module) and Perl documentation (for the `POD::Usage PERL` module) should be installed on your system. If not, you will get an error message when you try to use the `lcg-info` command. See section 2.1.2 on how to install necessary packages.

Table 3.3 – Options for the `lcg-info` command

<code>--help</code>	Prints the manual page and exits.
<code>--list-attrs</code>	Prints a list of the attributes that can be queried.
<code>--list-ce</code>	Lists the CEs <sub>?</sub> which satisfy a query, or all the CEs <sub>?</sub> if no query is given.
<code>--list-se</code>	Lists the SEs <sub>?</sub> which satisfy a query, or all the SEs <sub>?</sub> if no query is given.
<code>--query</code>	Restricts the output to the CEs <sub>?</sub> (SEs <sub>?</sub> ) which satisfy the given query.
<code>--bdii</code>	Allows to specify a BDII in the form <code>&lt;hostname&gt;:&lt;port&gt;</code> . If not given, the value of the environmental variable <code>LCG_GFAL_INFOSYS</code> is used. If that is not defined, the command returns an error.
<code>--sed</code>	Prints the output in a "sed-friendly" format: % separate the CE (SE) identifier and the printed attributes, & separate the values of multi-valued attributes.
<code>--quiet</code>	Suppresses warning messages.
<code>--attrs</code>	Specifies the attributes whose values should be printed.
<code>--vo</code>	Restricts the output to CEs <sub>?</sub> or SEs <sub>?</sub> where the given VO is authorized. Mandatory when VO-dependent attributes are queried upon.

To find out more information about the command, invoke it with the `--help` option. Use the following command:

```
$ lcg-info --help
```

To list the attributes supported by `lcg-info` command, use the following command line:

```
$ lcg-info --list-attrs
WaitingJobs      GlueCE           GlueCEStateWaitingJobs
MaxCPUTime       GlueCE           GlueCEPolicyMaxCPUTime
MaxTotalJobs     GlueCE           GlueCEPolicyMaxTotalJobs
CEStatus         GlueCE           GlueCEStateStatus
CEVOs            GlueCE           GlueCEAccessControlBaseRule
FreeCPUs         GlueCE           GlueCEStateFreeCPUs
RunningJobs      GlueCE           GlueCEStateRunningJobs
MaxWCTime        GlueCE           GlueCEPolicyMaxWallClockTime
Accesspoint      GlueCESEBind    GlueCESEBindCEAccesspoint
```

CloseCE	GlueCESEBindGroup	GlueCESEBindGroupCEUniqueID
CloseSE	GlueCESEBindGroup	GlueCESEBindGroupSEUniqueID
AvailableSpace	GlueSA	GlueSAStateAvailableSpace
Path	GlueSA	GlueSAPath
SEVos	GlueSA	GlueSAAccessControlBaseRule
UsedSpace	GlueSA	GlueSAStateUsedSpace
Root	GlueSA	GlueSARoot
SESite	GlueSE	GlueForeignKey
SEName	GlueSE	GlueSEName
SE	GlueSE	GlueSEUniqueID
SEType	GlueSE	GlueSEType
SEPort	GlueSE	GlueSEPort
SEArch	GlueSE	GlueSEArchitecture
Protocol	GlueSEAccessProtocol	GlueSEAccessProtocolType
ArchType	GlueSL	GlueSLArchitectureType
SMPSize	GlueSubCluster	GlueHostArchitectureSMPSize
Processor	GlueSubCluster	GlueHostProcessorModel
OS	GlueSubCluster	GlueHostOperatingSystemName
Cluster	GlueSubCluster	GlueSubClusterUniqueID
Tag	GlueSubCluster	GlueHostApplicationSoftwareRunTimeEnvironment
CINT2000	GlueSubCluster	GlueHostBenchmarkSI00
OSVersion	GlueSubCluster	GlueHostOperatingSystemVersion
CFP2000	GlueSubCluster	GlueHostBenchmarkSF00
OSRelease	GlueSubCluster	GlueHostOperatingSystemRelease
ClockSpeed	GlueSubCluster	GlueHostProcessorClockSpeed
Memory	GlueSubCluster	GlueHostMainMemoryRAMSize

For each attribute the simplified attribute name used by `lcg-info`, the corresponding object class and the GLUE attribute name are given [3.1, 3.2]<sup>2</sup>.

## Listing and querying information on CEs<sup>2</sup> with `lcg-info`

Let us try to find out list of computing elements available. For this we use the `--list-ce` option as given in the following example:

```
$ lcg-info --list-ce
```

The output is similar to the following:

```
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-infinite
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-long
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-short
- CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-egrid
- CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats
- CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-stormdev
- CE: grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-cert
- CE: grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-grid
- CE: grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-gridats
- CE: gridts01.grid.elettra.trieste.it:2119/jobmanager-lcgpbs-gridats
```

To find out list of computing elements only related to `gridats`, we need to specifically indicate the VO

using `--vo` option:

```
$ lcg-info --list-ce --vo gridats
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-infinite
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-long
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-short
- CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats
- CE: grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-gridats
- CE: gridts01.grid.elettra.trieste.it:2119/jobmanager-lcgpbs-gridats
```

Let us further extend our command to find out the total number of CPU (`TotalCPUs`) and free ones (`FreeCPUs`). The `--vo` attribute must be provided since we are querying for attributes (table 3.2). Make sure to put the attributes one after another separated by commas (,) within single-quotes ('). No spaces should be kept between attributes.

```
$ lcg-info --list-ce \
  --attrs 'TotalCPUs,FreeCPUs' --vo gridats
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-infinite
  - TotalCPUs      2
  - FreeCPUs       2
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-long
  - TotalCPUs      2
  - FreeCPUs       2
- CE: ce-02.grid.sissa.it:2119/jobmanager-lcgpbs-short
  - TotalCPUs      2
  - FreeCPUs       2
- CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats
  - TotalCPUs      6
  - FreeCPUs       5
- CE: grid001.oat.ts.astro.it:2119/jobmanager-lcgpbs-gridats
  - TotalCPUs     44
  - FreeCPUs     28
- CE: gridts01.grid.elettra.trieste.it:2119/jobmanager-lcgpbs-gridats
  - TotalCPUs     22
  - FreeCPUs     22
```

Let's make our command further complicated by trying to find out machines with exactly 6 CPUs:

```
$ lcg-info --list-ce \
  --query 'TotalCPUs=10' \
  --attrs 'TotalCPUs,FreeCPUs' \
  --vo gridats
- CE: egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-gridats
  - TotalCPUs      6
  - FreeCPUs       5
```

Note

*Note:*

It must be stressed that `lcg-info` command only supports a logical *AND* of logical expressions, separated by commas, and the allowed operators are `>=`, `<=` and `=`. The `*` symbol matches any number of characters.

Use the following command to locate any CE running "Debian" as the operating system and which is part of gridats; use the OS attribute to construct a query:

```
$ lcg-info --list-ce \  
  --query 'OS=Debian' --vo gridats  
- CE: ce-01.grid.sissa.it:2119/jobmanager-lcgpbs-gridats  
- CE: egrid.create-net.org:2119/jobmanager-lcgpbs-gridats
```

Use the OS and OSVersion attributes to list the OS name and version:

```
$ lcg-info --list-ce \  
  --query 'OS=Debian' \  
  --vo gridats \  
  --attrs 'OS,OSVersion'  
- CE: ce-01.grid.sissa.it:2119/jobmanager-lcgpbs-gridats  
  - OS Debian  
  - OSVersion 3  
- CE: egrid.create-net.org:2119/jobmanager-lcgpbs-gridats  
  - OS Debian  
  - OSVersion 3
```

## Listing and querying information on SEs with `lcg-info`

So far our focus was only on computing elements. Let us try to find out information about storage elements using `lcg-info` command. Use the following command to list SEs in gridats:

```
$ lcg-info --list-se --vo gridats  
  
- SE: baciuco.grid.sissa.it  
  
- SE: ce-01.grid.sissa.it  
  
- SE: egrid-ce-01.egrid.it  
  
- SE: egrid.create-net.org  
  
- SE: grid001.oat.ts.astro.it
```

The following command uses the `UsedSpace` and `AvailableSpace` attributes to find out used and free space in each SE:

```
$ lcg-info --list-se \  
  --attrs 'UsedSpace,AvailableSpace' \  
  --vo gridats  
- SE: egrid-ce-01.egrid.it  
  - UsedSpace          1054456  
  - AvailableSpace     13659804  
  
- SE: grid002.oat.ts.astro.it  
  - UsedSpace          874446272  
  - AvailableSpace     688004000  
  
- SE: gridts01.grid.elettra.trieste.it  
  - UsedSpace          7753952  
  - AvailableSpace     169452728
```

## Querying the local GRIS directly

The local GRISes<sup>2</sup> running on CEs<sup>2</sup> and SEs<sup>2</sup> reports both static and dynamic information on the characteristics and status of services. This information can be directly accessed using `ldapsearch` command.

In order to access the GRIS on specific Grid elements the host name and the TCP port where GRIS run must be known. The default port number is 2135.

As an example use the following command to access GRIS running on host `baciuco.grid.sissa.it`:

```
ldapsearch -x -h baciuco.grid.sissa.it -p 2135 -b "mds-vo-name=local,o=grid"
```

Where the `-x` option indicates that simple authentication to be used; the `-h` and `-p` options precede the host name and port respectively; and the `-b` option is used to specify the initial entry for the search in the LDAP tree. For a GRIS, the initial entry of the DIT is always `o=grid`, and the second one (next level) is `mds-vo-name=local`. It is in the entries in the deeper levels, that the actual resource information is shown. That is why `mds-vo-name=local,o=grid` is used as DN of the initial node for the search.

The result of the command can be *very* long. Therefore understanding hierarchy (objects, attributes and their relationships) will not be so easy. In order to restrict the search, filters can be used; the filters are applied on given attributes and conditions. The operator used in a filter can be one of the following:

- =  
Entries whose attribute is equal to the value
- >=  
Entries whose attribute is greater than or equal to the value
- <=  
Entries whose attribute is less than or equal to the value
- =\*  
Entries that have a value set for that attribute
- ~=  
Entries whose string attribute value *approximately* matches the specified value; the SOUNDEX algorithm is used for comparison.

Complex search filters can be formed by using Boolean operators to combine constraints. The Boolean operators that can be used are: AND (&), OR (|) and NOT (!).

## Using a graphical LDAP Browser

However to make things simple and clear we can make use of a graphically LDAP browser. There are various commercial as well as Open Source LDAP client utilities. Those provide a clear and hierarchical representation of data. Searching and navigating (or even editing of values) with such tools is really easy. For rest of the examples we will be using the *LDAP BrowserEditor*<sup>2</sup> [3.7]<sup>2</sup> by Jarek Gawor which is free for educational user.

## Installing the *LDAP BrowserEditor*<sup>2</sup>

Installing *LDAP BrowserEditor*<sup>2</sup> is really simple however you must have Java 1.2 or higher installed on your machine (if not see section 2.1.3 on installing Java).

### Step 1

Download the LDAP BrowserEditor<sub>2</sub> from:

<http://www-unix.mcs.anl.gov/~gawor/ldap/>

Download the latest Browser<version>.tar.gz file.

### Step 2

Uncompress and extract archive:

```
$ gunzip Browser"version".tar.gz
```

It will be extracted to a new directory called ldapbrowser. You may move the ldapbrowser directory to a different location if needed.

If the ldapbrowser directory is created and if it is not empty the installation is complete.

## Running the *LDAP BrowserEditor<sub>2</sub>*

### Step 1

To execute the *LDAP BrowserEditor<sub>2</sub>* use the following commands. Before running change the directory to ldapbrowser, then execute the lbe.sh script as follows:

```
$ cd ldapbrowser
$ ./lbe.sh
```

Then the Connect dialog box appears (figure 3.4). This dialog box can be used to create and manage connections to LDAP servers which are referred as sessions. By default it has a connection to LDAP server running in University of Michigan (ldap.itd.umich.edu).

Figure 3.4 – The Connect dialog box

### Step 2

Select *uofmichigan* from the *Session List*: list box then, click the Edit Button. You will observe the configuration parameters for *uofmichigan* session and how those are given (figure 3.5).

Close the dialog by clicking the Close button.

Figure 3.5 – The Edit Session dialog box

### Step 3

Then click the *Connect* button on the *Connect* dialog box. After couple of seconds you will be connected to the University of Michigan LDAP server. You may not be able to browse any values however use the main window to identify different components (figure 3.6).

Figure 3.6 – The main window of the *LDAP BrowserEditor<sub>2</sub>*

### Step 4

Close the connection by selecting *File -> Disconnect* from the menu.

## Browsing Grid Resources with LDAP Browser

### Step 1

Let us connect to the *gridts04.grid.elettra.trieste.it* (is a CE having some SE capabilities as well) and observe its parameters.

To reopen the Connection dialog back click *File* → *Connect* menu or the shortcut key *Ctrl+C*. Then the connection dialog box should appear.

*Step 2*

Click *New* button to create a new connection. Then the *New Session* dialog box appears. In the *New Session* dialog box, click the *Name* tab. Then enter `gridts04` in the text box labelled *Name*:

Select the *Connection* tab. Enter the following values in the *Host Info* pane:

Host: `gridts04.grid.elettra.trieste.it`

Port: `2170`

Base DN: `mds-vo-name=local, o=grid`

..note:: *Note:*

You do not need to provide anything under *User Info* pane since EGEE base Grids allow anonymous access to the catalog.

*Step 3*

Then click the *Save* button to save the new parameters.

Figure 3.7 – Left pane in *LDAP Browser/Editor* main window

Figure 3.8 – The main window indicating objects and corresponding attributes

*Step 4*

Select `gridts04` from the list box labelled *Session List*: and click *Connect* button.

After few seconds you will see the DIT of `gridts04` (figure 3.7) appearing on the left of the main window. The LDAP objects are displayed in the form of a tree and all attributes of the entries in the form of a table (right of the window called the attributes pane).

*Step 5*

Browse through other objects and attributes listed. Try to find out the answer to the following questions by browsing the various object and their attributes.

1. Grid sites that `gridts04` belongs to?
2. Number of CPUs??
3. Number of free CPUs??
4. CPU vendor?
5. Amount of RAM installed?

*Step 6*

Locating certain attributes can take a while when there is lots of information on the DIT. You can use the search facility to find out values of specific attributes. In order to do so, you must know the GLUE attribute name.

Suppose you want to find the amount of memory. The GLUE attribute used to hold memory (RAM size) is `GlueHostMainMemoryRAMSize`.

Select the top most object from the DIT. Then select *View* → *Search* from the menu. The *Search* dialog box appears (figure 3.9).

Figure 3.9 – The Search dialog box

Type `GlueHostMainMemoryRAMSize?` in the text box labelled *Attributes:*.

Then click the Search button.

By default it will search attributes only at one-level. If the search need to be performed even on sub trees select the Sub-tree level radio button under the *Search* scope:.

Figure 3.10 illustrates an output of the above search.

Figure 3.10 – Results of a search query

Results of the search operation depend on the current object that is selected on the DIT. So if the desired result is not available try selecting a possible object from the DIT. A short cut could be to start with the DN (`mdu-v0-name=local,o=grid`) which is at the top of the DIT. Then it will search for all sub-trees under it.

*Step 7*

Try to find out information on the following attributes. Also, note the objects and DNs of the objects that they belong to.

`GlueCEInfoTotalCPUs?`

`GlueCEStateFreeCPUs?`

`GlueSAStateAvailableSpace?`

`GlueHostOperatingSystemName?`

`GlueHostOperatingSystemVersion?`

`GlueHostProcessorClockSpeed?`

*Step 8*

Rather than querying individual site GIISes?, site BDIIs or local GRISes? of CEs? and SEs?, we can also query the top most BDII.

Let us try to connect to the top most BDII for `gridats` which is accessible through port 2170 of `gridts04.grid.elettra.trieste.it`.

Use the *LDAP BrowserEditor?* and create a new session called *Top BDII*. Then connect to the top most BDII and browse its objects and parameters. Look for various CEs? and SEs? under it.