



Enabling Grids for E-science

Hands-on on R-GMA

Tony Calanducci

INFN Catania

gLite tutorial at the EGEE User Forum

Geneva, 27-28 February 2006

www.eu-egee.org



- To start the R-GMA command line tool, run the following command:

>rgma

- On startup, you should receive the following message:

```
Welcome to the R-GMA virtual database for Virtual
Organisations.

=====
Your local R-GMA server is:
  https://rgmasrv.ct.infn.it:8443/R-GMA
You are connected to the following R-GMA Registry services:
  https://rgmasrv.ct.infn.it:8443/R-GMA/RegistryServlet
You are connected to the following R-GMA Schema service:
  https://rgmasrv.ct.infn.it:8443/R-GMA/SchemaServlet
Type "help" for a list of commands.
rgma>
```

- **Commands** are entered by typing at the **rgma>** prompt and hitting 'enter' to execute the command.
- A **history** of the commands executed can be accessed using the Up and Down arrow keys.
- To **search a command from history** use CTRL-R and type the first few letters of the command to recall.
- **Command autocompletion** is supported (use Tab when you have partly entered a command).

General Commands

- **help**

Display general help information.

- **help <command>**

Display help for a specific command.

- **exit or quit**

Exit from R-GMA command line interface.

- **show tables**

Display the name of all tables existing in the Schema

- **describe <tablename>**

Show all information about the structure of a table

- Querying data uses the standard SQL SELECT statement, e.g.:

```
rgma> SELECT * FROM GlueService
```

The behaviour of SELECT varies according to the type of query being executed. In R-GMA there are three basic types of query:

- **LATEST Queries** only the most recent tuple for each primary key
- **HISTORY Queries** all historical tuples for each primary key
- **CONTINUOUS Queries** returns tuples continuously as they are inserted.

- The type of query can be changed using the **SET QUERY** command as follow:

```
rgma> SET QUERY LATEST
```

or

```
rgma> SET QUERY CONTINUOUS
```

- The current query type can be displayed using **rgma> SHOW QUERY**

1. Display all the table of the Schema

```
rgma>show tables
```

2. Display information about GlueSite table

```
rgma>describe GlueSite
```

3. Basic select query on the table named GlueSite

```
rgma>set query latest
```

```
rgma>show query
```

```
rgma>select Name,Latitude,Longitude from  
GlueSite
```

- The maximum age of tuples to return can also be controlled. To limit the age of latest or historical tuples use the **SET MAXAGE** command. The following are equivalent:

```
rgma> SET MAXAGE 2 minutes
```

```
rgma> SET MAXAGE 120
```

- The current maximum tuple age can be displayed using **rgma> SHOW MAXAGE**
- To disable the maximum age, set it to none:

```
rgma> SET MAXAGE none
```

- **The final property affecting queries is timeout.**
 - For a latest or history query the timeout exists to prevent a problem (e.g. network failure) from stopping the query from completing.
 - For a continuous query, timeout indicates how long the query will continue to return new tuples. Default timeout is 1 minute and it can be changed using

rgma>SET TIMEOUT 3 minutes or SET TIMEOUT 180

- **The current timeout can be displayed using**
rgma>SHOW TIMEOUT

- The SQL INSERT statement may be used to add data to the system:

```
rgma> INSERT INTO userTable VALUES ('a', 'b', 'c', 'd')
```

- In R-GMA, data is inserted into the system using a **Producer** component which handles the INSERT statement.
- Using the command line tool you may work with one producer at a time.
- The current producer type can be displayed using:

```
rgma>show producer
```
- The producer type can be set using:

```
rgma>set producer latest
```

1. Publish and consume tuple with continuous Producers

```
rgma>describe userTable
```

```
rgma>set producer continuous
```

```
rgma>insert into userTable values('cod','string',1.4,66)
```

```
rgma>set query continuous
```

```
rgma>set maxage 1 minutes
```

```
rgma>set timeout 10 seconds
```

```
rgma>select * from userTable
```

- To instruct the secondary producer to consume from table MyTable:

```
rgma> SECONDARYPRODUCER userTable
```

- Like the producer, the secondary producer may be configured to answer latest and/or history queries:

```
rgma> SET SECONDARYPRODUCER latest
```

- (By default the secondary producer can answer both latest and history queries.)
- The current secondary producer type can be displayed using:

```
rgma> SHOW SECONDARYPRODUCER
```

2. Consume tuples coming from secondary producers:

```
rgma>set secondaryproducer latest
```

```
rgma>secondaryproducer userTable
```

```
rgma>show producers of userTable
```

```
rgma>set producer continuous
```

```
rgma>insert into userTable values ('cod','string',5.2,44)
```

```
rgma>set query latest
```

```
rgma>select * from userTable
```

