EGEE

R-GMA Server

Document identifier:	**EGEE-JRA1-TEC-server-servlet-manual**
Date:	**August 17, 2006**
Activity:	**JRA1: Middleware Engineering and Integration (UK Cluster)**
Document status:	**DRAFT**
Document link:	

Abstract: This document describes the servlet-based implementation of the R-GMA server

CONTENTS

1 INTRODUCTION

The R-GMA server is a Java servlet-based web application which provides the Consumer, Producer, Registry and Schema services for the R-GMA distributed information and monitoring system.

The server is designed to be run within a servlet container such as Jakarta Tomcat. Tomcat versions 4 and 5 have been tested, however other versions or other servlet containers may also work.

2 DEPLOYMENT

2.1 INSTALLING THE PACKAGES

The following packages are required to run the R-GMA server:

```
glite-rgma-base
glite-rgma-server-servlet
```

If you are deploying an Schema service, you are recommended to also install the package:

```
glite-rgma-standard-tables
```

This will ensure that a standard set of tables will be available to R-GMA clients without having to create them manually. The service and site publishers as well as some of the tests rely on these tables being available.

Finally, you must then set the environment variable RGMA_HOME to the prefix of the R-GMA installation, for example:

```
export RGMA_HOME=/opt/glite
```

2.2 CONFIGURATION OF THE SERVER

2.2.1 WHICH SERVICES TO DEPLOY?

An R-GMA server may run any combination of the following sets of services:

- The Consumer and Producer services

  These are the basic R-GMA services that allow data to be inserted and retrieved from the system. Normally these are the only services an R-GMA client connects to.

- The Registry service

  This service keeps track of Producers and Consumers in existance and provides information about which Producers are relevant to which Consumer queries.

- The Schema service

  This service provides information about tables that have been defined, their names and their columns. A table may not be used until it has been defined in the Schema.

- The Browser service

  This service provides a simple web-browser based interface to R-GMA.

Normally only one of first three sets will be deployed at once, i.e. a server will be a Schema, or a Registry, or a Consumer/Producer server. However this is not compulsary. The Browser service can be run on any server.

Currently, in an R-GMA deployment there is only one Schema server. The Registry service may be replicated so there may be more than one Registry server, however there are expected to be a small number of Registry servers compared to Consumer/Producer servers. If you are running Registry replication, i.e. there is more than one Registry server, ensure that all of the Registry servers know about each other.

If you are setting up a new R-GMA deployment you will need to select one server to run the Schema, one or more servers to run Registry services, and run Consumer/Producer services on the others. If you are joining an existing R-GMA deployment it is likely that you will want to run only the Consumer/Producer services and connect to an existing Schema and Registry replica.

2.2.2 CLIENT CONFIGURATION

The R-GMA server must also act as an R-GMA client when connecting to other R-GMA servers. To configure the client-side of the system, run:

```
$RGMA_HOME/bin/rgma-setup.py
```

This script will prompt [1] you to enter the endpoints of all the R-GMA services. The default for each is the local machine. If you are running the Consumer/Producer services you should ensure that the endpoint for these services matches the local hostname.

If you are running a Registry services you should ensure that the Registry endpoint matches the local hostname and enter the endpoints of any Registry replicas when prompted. Otherwise, set the endpoints of all the Registry replicas in the deployment.

If you are running a Schema service you should ensure that the Schema endpoint matches the local hostname. Otherwise set this to the endpoint of the server that is running the Schema service.

2.2.3 SERVER CONFIGURATION

The R-GMA server is configured by running:

```
$RGMA_HOME/share/rgma/script/rgma-server-setup.py
```

Appendix A describes the full usage of this command, but the following examples may be useful:

To configure a machine to run the Consumer/Producer services using a MySQL database running on the local machine:

```
$RGMA_HOME/bin/rgma-server-setup.py
```

To configure a machine to run the Consumer/Producer services using a remote MySQL database:

```
$RGMA_HOME/share/rgma/scripts/rgma-server-setup.py
    --producerdb=jdbc:mysql://<DB host URL>:3306/
    --produceruser=<DB username>
    --producerpass=<DB password>
```

---

[1]The script can also be run non-interactively for automated configuration. Use the –help option for more details

To configure a machine to run the Schema and Browser service using a local MySQL database with non-default username and password:

```
$RGMA_HOME/share/rgma/scripts/rgma-server-setup.py --schema=yes --browser=yes
    --schemauser=<DB username>
    --schemapass=<DB password>
```

To configure a machine to run the Registry service using a remote MySQL database:

```
$RGMA_HOME/share/rgma/scripts/rgma-server-setup.py --registry=yes
    --registrydb=jdbc:mysql://<DB host URL>:3306/registry
    --registryuser=<DB username>
    --registrypass=<DB password>
```

This configuration script outputs a file containing SQL commands to set up the required databases and access permissions on a MySQL database. You should pipe this file into your MySQL database as root.

2.2.4 CONFIGURATION FILES

The R-GMA client configuration file is:

```
$RGMA_HOME/etc/rgma/rgma.conf
```

It is a simple name=value properties file which specifies the endpoints of the R-GMA services.

The server configuration file is:

```
$RGMA_HOME/etc/rgma-server/rgma-server.conf
```

It is an XML file and contains various parameters which may be used to control the behaviour of the server. Appendix B describes the content of this file in detail.

2.3 INSTALLING THE WEB APPLICATION

The web application is installed by copying the file `R-GMA.war` into the `webapps` directory of the servlet container, e.g.

```
cp $RGMA_HOME/share/webapps/R-GMA.war /var/tomcat4/webapps/
```

If there is an existing `R-GMA` directory in the `webapps` directory, it should be removed. The servlet container should then be restarted.

3 THE BROWSER SERVLET

3.1 ACCESSING THE BROWSER

The browser servlet allows access to R-GMA from any web browser. To access the browser, point your web browser to the URL:

```
https://<host running the browser>:8443/R-GMA/
```

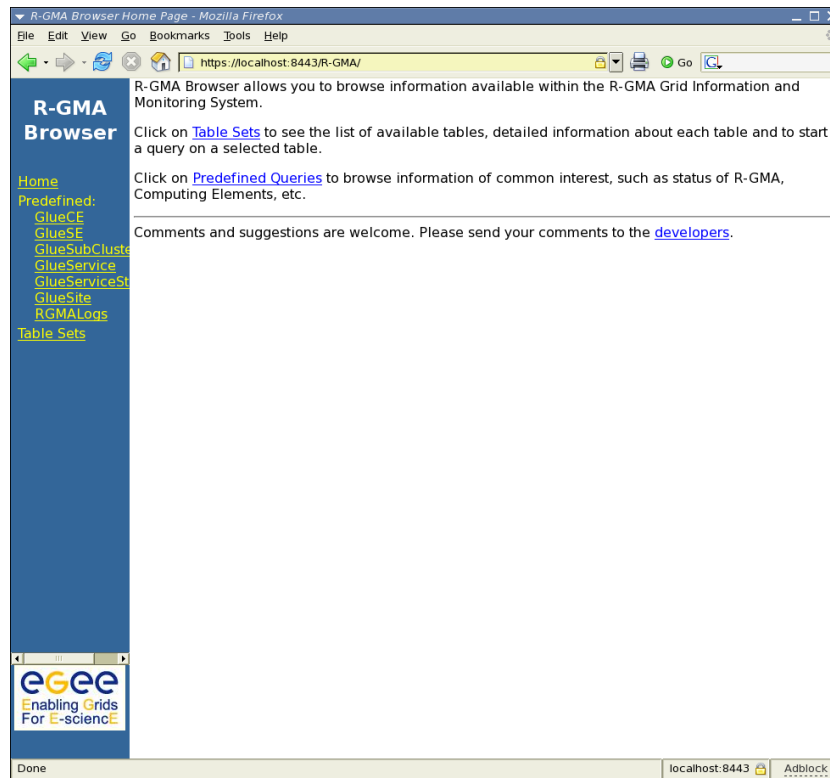Figure 1 shows the home page of the browser.

Figure 1: The R-GMA browser

3.2 ISSUING QUERIES

Click on the Table Sets link on the home page, and click on the table to query. A form similar to Figure 2 will be displayed.

The list of fields at the top of the page controls which columns will be included in the query. The default is for all columns to be included. The text block below is used to enter an optional `WHERE` clause, for example `GlueSite_UniqueId='heplnw12.pp.rl.ac.uk'`.

The radio buttons below this select the type of query (Static, History, Latest or Continuous). If the 'Old' checkbox is ticked for a continuous query it will also return tuples published in the past that have not been expired from the Continuous Producer. For continuous queries, the timeout gives the length of time to run the query for in seconds. For static, latest and history queries this gives the length of time to wait for results to arrive before aborting the query.

Finally, click on the Query button to issue the query. The results will be displayed in a table (Figure 3).

3.3 DIRECTED QUERIES

Normally you will want to use the R-GMA mediator to automatically select which Producers to query for information. However if you want to specify the producers manually, choose the 'Select producers you want to query' option. Then select producers from the list below. This is known as a 'directed query'. You must ensure that you select Producers that are able to answer the type of query you are posing or no results will be returned.
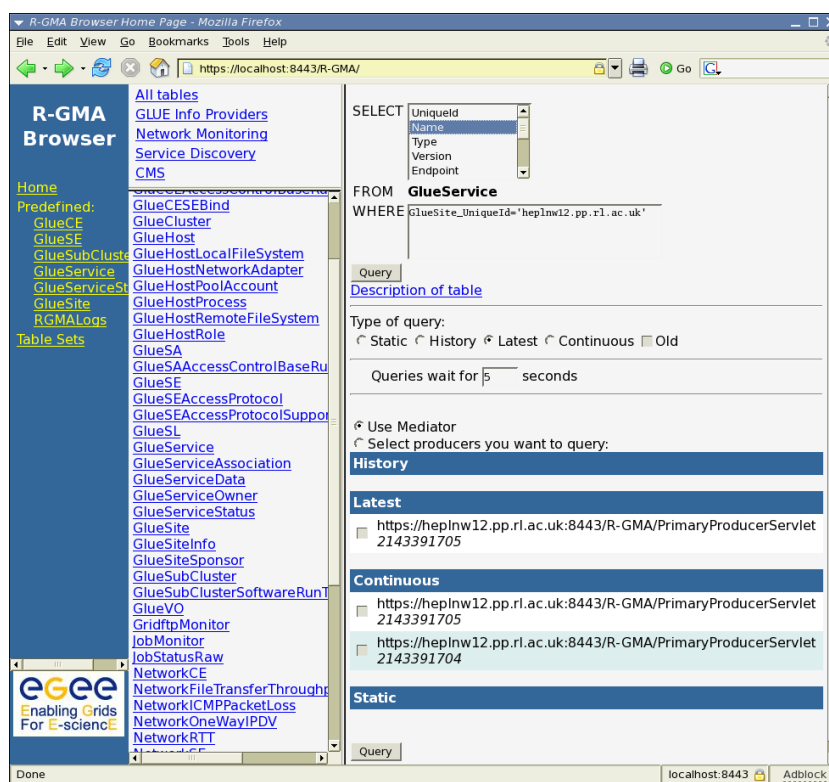
R-GMA SERVER

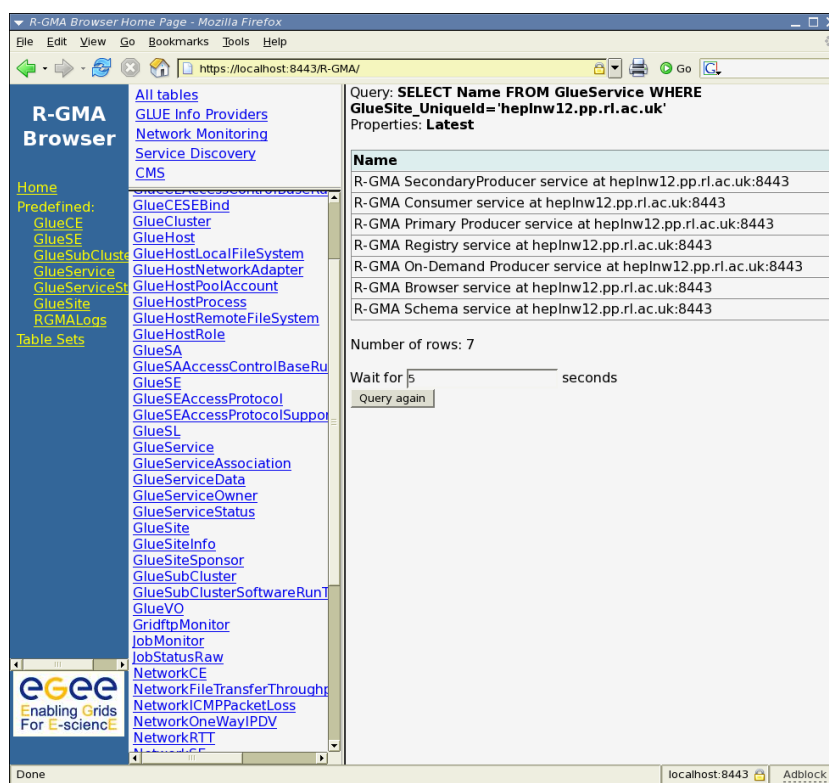**Figure 2:** Issuing a query

Figure 3: The results of a query

3.4 TABLE DESCRIPTION

To view a table schema, from the table query page (Figure 2), click the 'Description of Table' link, below the query button.

4 LOGGING INFORMATION

The R-GMA server writes logging information to the file:

```
/var/log/glite/rgma-server/rgma-server.log
```

Logging may be customised by editing the log4j.properties file for the server. The location of this file is:

```
$RGMA_HOME/etc/rgma-server/log4j.properties
```

It is necessary to restart the server in order to apply changes to the logging configuration. For a full description of the format of this file see the log4j documentation.

4.1 CHANGING THE GLOBAL LOGGING LEVEL

To change the severity of messages to be reported in the R-GMA log file, change WARN in the line:

```
log4j.rootLogger=WARN, A1
```

to some other value. Recognised values are NONE, FATAL, ERROR, WARN, INFO, DEBUG, and ALL (decreasing severity, i.e. increasing verbosity). Be aware that setting the log level to DEBUG will results in a very high density of logging messages.

4.2 CHANGING THE LOGGING LEVEL FOR A SPECIFIC CLASS

To increase the logging information for a particular class, uncomment the line which refers to the class and change the logging level if required. For example, changing:

```
#log4j.logger.org.edg.info.RegistryInstance=DEBUG
```

to

```
log4j.logger.org.edg.info.RegistryInstance=DEBUG
```

will enable debug logging for the RegistryInstance class. This is useful if you are interested in detailed information about one part of the system but do not want to see a large number of messages from other parts.

4.3 CHANGING THE SIZE OF THE LOG FILE

By default the log file has a maximum size of 10Mb and the last 10 files are kept (i.e. a maximum space usage of 100Mb). To control this, edit the following lines:

```
log4j.appender.A1.MaxFileSize=20MB
log4j.appender.A1.MaxBackupIndex=10
```

It is not recommended to change the appender type to a simple FileAppender as this may result in the log file growing uncontrollably and consuming excess disk space.

5 HOST-BASED ACCESS CONTROL

The R-GMA server has a host-based access control mechanism that can be used to allow and deny access to the services using a flexible regular-expression based system. The access control list is contained in the file:

```
$RGMA_HOME/etc/rgma-server/access-control-list.xml
```

The file is in XML format and has three parts which may be modified:

- The list of hosts that are allowed access

- The list of hosts that are denied access

- The order in which to evaluate the allowed and denied lists

5.1 THE ORDER PARAMETER

The order in which the allow and deny lists are evaluated is controlled by the `order` attribute of the root `<access-list>` element. It must consist of the words `allow` and `deny` separated either by spaces or a comma. The words are case insensitive.

5.1.1 EXAMPLES

```
<access-list order="allow,deny">     OK
<access-list order="deny allow">     OK
<access-list order="Deny , Allow">   OK
<access-list>                        OK, uses default allow, deny order
<access-list order="deny">           ERROR, must contain allow and deny
<access-list order="allow:deny">     ERROR, wrong separator
```

If the order `allow,deny` is specified, the `allow` list is evaluated first and access is turned on for all matching hosts. The `deny` list is then evaluated and access is turned off for all matching hosts. Any host not matching an entry in either list is denied access.

If the order `deny,allow` is specified, the `deny` list is evaluated first and access is turned off for all matching hosts. The `allow` list is then evaluated and access is turned on for all matching hosts. Any host not matching an entry in either list is denied access.

In general the `allow,deny` order is more secure because non-matching hosts are denied access by default. It also makes it easy to allow access to groups of hosts but to deny specific hosts within these groups. If the `order` attribute is not specified or is not of the required form, this is used as the default.

5.2 THE ALLOW AND DENY LISTS

The allow and deny lists contain any number of `<host>` elements, with the format:

```
<host servlet="servlet specifier">host regular expression</host>
```

5.2.1 THE SERVLET SPECIFIER

The servlet specifier is the name of the servlet to allow or deny access to. Recognised servlets are:

```
PrimaryProducerServlet
SecondaryProducerServlet
OnDemandProducerServlet
LatestProducerServlet
StreamProducerServlet
DataBaseProducerServlet
CanonicalProducerServlet
ConsumerServlet
ArchiverServlet
RegistryServlet
SchemaServlet
```

The wildcard `*` can be used to mean 'all servlets'. Note that the LatestProducer, StreamProducer, DataBaseProducer and CanonicalProducerServlet are deprecated however are still present for compatibility. The authorisation rules for the first three should be the same as for the PrimaryProducer, the Archiver should match the SecondaryProducer, and the CanonicalProducer should match the OnDemandProducer.

5.2.2 THE HOST REGULAR EXPRESSION

The host regular expression should be a regular expression compatible with the Java Pattern class [2]. The allow/deny rule will be applied to any host matching the regular expression, for example:

5.2.3 EXAMPLES

Match all hosts accessing the Registry servlet

```
<host servlet="RegistryServlet">.*</host>
```

Match all hosts at the site `.cern.ch` accessing the Schema servlet

```
<host servlet="SchemaServlet">.*\.cern\.ch$</host>
```

Match the host `lcgic03.gridpp.rl.ac.uk` accessing any servlet

```
<host servlet="*">^lcgic03\.gridpp\.rl\.ac\.uk$</host>
```

---

[2]These are very similar to Perl regular expressions

5.3 COMPLETE EXAMPLES

Allow only hosts from the sites `.cern.ch` and `.gridpp.rl.ac.uk`, and block a single host from accessing any servlet:

```
<access-list order="allow, deny">

  <hosts-allow>
    <host servlet="*">.*\.cern\.ch$</host>
    <host servlet="*">.*\.gridpp\.rl\.ac\.uk$</host>
  </hosts-allow>

  <hosts-deny>
    <host servlet="*">^lxb2001\.cern\.ch$</host>
  </hosts-deny>

</access-list>
```

Allow any host to access the Registry and Schema servlets, but only one specific host to access other servlets:

```
<access-list order="allow, deny">

  <hosts-allow>
    <host servlet="RegistryServlet">.*</host>
    <host servlet="SchemaServlet">.*</host>
    <host servlet="*">^lcg34\.gridpp\.rl\.ac\.uk$</host>
  </hosts-allow>

  <hosts-deny>
  </hosts-deny>

</access-list>
```

Allow access to any host except for two blocked hosts (uses `deny,allow` order):

```
<access-list order="deny, allow">

  <hosts-allow>
  </hosts-allow>

  <hosts-deny>
    <host servlet="*">^lcg49\.gridpp\.rl\.ac\.uk$</host>
    <host servlet="*">^lxb2004\.cern\.ch$</host>
  </hosts-deny>

</access-list>
```

A SERVER SETUP SCRIPT REFERENCE

The R-GMA server setup script is

`$RGMA_HOME/share/rgma/script/rgma-server-setup.py`

It performs the following tasks:

- Writes the configuration file for the server

- If a Schema is being deployed and the `glite-rgma-standard-tables` package is installed, processes schema files describing the standard tables so they will be available at startup

- Generates SQL commands to set up databases used by the server

- If the R-GMA service tool is installed, arranges for the local R-GMA services to be published to the Service and ServiceStatus tables

A.1 PARAMETERS

The setup script takes various parameters, all of the form `--<name>=<value>`.

Parameter	Description	Default value
`--schema`	Whether to deploy the Schema service. Should be `yes` or `no`	`no`
`--registry`	Whether to deploy the Registry service. Should be `yes` or `no`	`no`
`--browser`	Whether to deploy the Browser service. Should be `yes` or `no`	`no`
`--producerdb`	JDBC url of database where Producer databases will be created	`jdbc:mysql://localhost:3306/`
`--produceruser`	Username to use when accessing the Producer databases	`producer`
`--producerpass`	Password to use when accessing the Producer databases	`rgma`
`--registrydb`	JDBC url of the Registry database	`jdbc:mysql://localhost:3306/registry`
`--registryuser`	Username to use when accessing the Registry database	`registry`
`--registrypass`	Password to use when accessing the Registry database	`rgma`
`--schemadb`	JDBC url of the Schema database	`jdbc:mysql://localhost:3306/schema`
`--schemauser`	Username to use when accessing the Schema database	`schema`
`--schemapass`	Password to use when accessing the Schema database	`rgma`
`--port`	Port number the server will be run on. This is used to construct the service endpoints for the service tool configuration files.	`8443`
`--output`	Output file for SQL commands	`rgma-setup-db.sql`

A.2 OPTIONS

The following options are supported by the setup script:

Option	Description
--help	Displays usage information
--version	Displays version number

B CONFIGURATION FILE REFERENCE

The R-GMA server configuration file is an XML file with the following format:

```
<RGMA-Server>
  <Common>
    <!-- Parameters shared by more than one service -->
  </Common>

  <Producer>
    <!-- Parameters used by the Producer services -->
  </Producer>

  <Consumer>
    <!-- Parameters used by the Consumer service -->
  </Consumer>

  <Registry>
    <!-- Parameters used by the Registry service -->
  </Regsitry>

  <Schema>
    <!-- Parameters used by the Schema service -->
  </Schema>

  <Browser>
    <!-- Parameters used by the Browser service -->
  </Browser>

</RGMA-Server>
```

In general, the absence of a section from the configuration file means that the corresponding service will not be deployed. Hence, the setup script only inserts those sections corresponding to services that were specified to be deployed.

Normally it will not be necessary to edit this file directly - the R-GMA server setup script configures the most important parameters. This appendix describes all the parameters available in the event that it is necessary to customise the server behaviour.

B.1 COMMON PARAMETERS

The Common section has the following format:

DRAFT

```
<Common>
  <accessListReadPeriod>60000</accessListReadPeriod>
  <connectionIdFile>WEB-INF/tmp/arch-con-id</connectionIdFile>
  <minTermInterval>600</minTermInterval>
  <maxTermInterval>129600</maxTermInterval>
</Common>
```

`accessListReadPeriod`	How often to re-read the access control list, in milliseconds
`connectionIdFile`	File to store connection IDs for R-GMA resources. This ensures that the same IDs are not re-used when an R-GMA server is restarted.
`minTerminationInterval`	Minimum termination interval for a Producer or Consumer resource, in seconds
`maxTerminationInterval`	Maximum termination interval for a Producer or Consumer resource, in seconds

B.2 PRODUCER PARAMETERS

The Producer section has the following format:

```
<Producer>
  <defaultMaxBufferSize>50000</defaultMaxBufferSize>
  <defaultMinRetentionPeriod>5</defaultMinRetentionPeriod>
  <cleanupPeriodForMinRetentionTracker>30</cleanupPeriodForMinRetentionTracker>
  <maxTupleMemory>35</maxTupleMemory>
  <minMemoryNeededForInsert>0</minMemoryNeededForInsert>
  <tableAlreadyExistsErrorCode>1050</tableAlreadyExistsErrorCode>
  <dbDriver>org.gjt.mm.mysql.Driver</dbDriver>
  <defaultDbLocation>jdbc:mysql://localhost:3306/ProducerDB_</defaultDbLocation>
  <defaultDbUsername>producer</defaultDbUsername>
  <defaultDbPassword>rgma</defaultDbPassword>
  <dbAdminLocation>jdbc:mysql://localhost:3306/mysql</dbAdminLocation>
  <dbAdminUsername>producer</dbAdminUsername>
  <dbAdminPassword>rgma</dbAdminPassword>
  <maxBlockSendSize>5000</maxBlockSendSize>
</Producer>
```

`defaultMaxBufferSize`	Maximum number of tuples allowed in the queue waiting to be streamed to Consumers
`defaultMinRetentionPeriod`	Default minimum retention period in minutes
`cleanupPeriodForMinRetentionTracker`	How often to check for expired tuples, in seconds
`maxTupleMemory`	Maximum amount of memory to be used for tuple storage, expressed as a percentage of the VM size. If set to -1 the amount of memory will be unlimited (until the VM runs out of memory).
`minMemoryNeededForInsert`	If greater than zero, insert statements will be blocked if the amount of free memory is less than this value, in Mb [3]
`tableAlreadyExistsErrorCode`	The error code returned by the database to indicate that the table in a `CREATE TABLE` statement already exists
`dbDriver`	The JDBC driver to use for Producer databases. Currently only MySQL databases are supported
`defaultDbLocation`	The JDBC URL to use for Producer databases when a specific database is not specified by the user
`defaultDbUsername`	The username to use for default Producer databases
`defaultDbPassword`	The password to use for default Producer databases
`dbAdminLocation`	The JDBC URL to use for admin-level operations on the Producer databases (for example creating a new Producer database)
`dbAdminUsername`	The username to use for admin database operations [4]
`dbAdminPassword`	The password to use for admin database operations
`maxBlockSendSize`	The maximum number of tuples to send to a Consumer in each streaming block

B.3 CONSUMER PARAMETERS

The Consumer section has the following format:

```
<Consumer>
  <bufferSize>10000</bufferSize>
  <streamingPort>8088</streamingPort>
  <notifyInterval>10000</notifyInterval>
  <defaultQueryTimeOutPeriod>30000</defaultQueryTimeOutPeriod>
  <responseTesterTimeOutPeriod>30000</responseTesterTimeOutPeriod>
</Consumer>
```

---

[4]This parameter currently has no effect. The minimum amount of memory is set to zero

[4]Normally the admin database details are the same as for the default database. The R-GMA server setup script configures MySQL to allow the Producer database user to create databases for Producers.

`bufferSize`	Maximum number of tuples that will be held by the Consumer servlet while waiting for the client to retrieve them via the `pop` method
`streamingPort`	The port number used for streaming ResultSets from Producers
`notifyInterval`	How often in milliseconds the Consumer will attempt to resend failed requests to Producers
`defaultQueryTimeOutPeriod`	The query timeout to use if one is not specified by the user, in milliseconds [5]
`responseTesterTimeOutPeriod`	How long in milliseconds to wait before timing out when testing which SecondaryProducer to use to answer a query

B.4 REGISTRY PARAMETERS

The Registry section has the following format:

```
<Registry replicate="true">
  <dbDriver>org.gjt.mm.mysql.Driver</dbDriver>
  <dbLocation>jdbc:mysql://localhost:3306/registry</dbLocation>
  <dbUsername>registry</dbUsername>
  <dbPassword>rgma</dbPassword>
  <replicationPeriod>30000</replicationPeriod>
  <notifyInterval>30000</notifyInterval>
  <duplicateKeyErrorCode>1062</duplicateKeyErrorCode>
</Registry>
```

`replicate`	Whether to enable registry replication. Should be `true` if more than one Registry is configured, `false` otherwise.
`dbDriver`	JDBC driver for the Registry database. Currently only MySQL databases are supported
`dbLocation`	JDBC URL of the Registry database.
`dbUsername`	Username to use when accessing the Registry database
`dbPassword`	Password to use when accessing the Registry database
`replicationPeriod`	Interval between replication cycles in milliseconds
`notifyInterval`	Interval in milliseconds to wait before retrying failed messages to Consumers and Producers
`duplicateKeyErrorCode`	Error code returned by the database to indicate a duplicate primary key

B.5 SCHEMA PARAMETERS

The Schema section has the following format:

```
<Schema>
  <dbDriver>org.gjt.mm.mysql.Driver</dbDriver>
  <dbLocation>jdbc:mysql://localhost:3306/schema</dbLocation>
  <dbUsername>schema</dbUsername>
  <dbPassword>rgma</dbPassword>
</Schema>
```

---

[5]This parameter currently has no effect. If the timeout is not specified, the query does not time out

dbDriver	JDBC driver for the Schema database. Currently only MySQL databases are supported
dbLocation	JDBC URL of the Schema database.
dbUsername	Username to use when accessing the Schema database
dbPassword	Password to use when accessing the Schema database

B.6 BROWSER PARAMETERS

The Browser has no parameters. However, the presence of this section determines if the browser service is available or not.

C KNOWN PROBLEMS AND CAVEATS

There is a know issue running Java 1.4.2 with the Scientific Linux 3 smp kernel on dual processor machines. The Java Virtual Machine crashes without reporting any errors.

A work around is to use a non-SMP kernel. There should be a fix for this bug in release 1.2.4_08 of Java. We also believe that the bug does not occur with Redhat 7.3 or Redhat Enterprise 4.