Practice: Executing OQL Queries

In this lab, you will instantiate couple of GemFire servers, and use a GemFire client application to populate and modify data in the servers. Then, you will use the GemFire DataBrowser tool to execute OQL queries on the data residing in the servers.

**What you will learn:**

* Start the GemFire members (servers and locator) by using gfsh utility.
* Execute a Java GemFire application client to add data.
* Launch GemFire DataBrowser tool, and execute OQL queries on the data.

.

### Start GemFire Locator by using gfsh

1. Start a Command prompt window, and add the example class directories to your CLASSPATH (if not done as a part of the earlier practice).

set CLASSPATH=  
<product\_directory>\ SampleCode\tutorial\classes;

<product\_directory>\SampleCode\helloworld\classes;

<product\_directory>\SampleCode\quickstart\classes;

<product\_directory>\SampleCode\examples\dist\classes;

<product\_directory>\lib\gemfire.jar;

%CLASSPATH%

where <product\_directory> corresponds to the location where you installed GemFire.

**IMPORTANT**: Make sure that for each Command prompt window, you need to set the required PATH as mentioned in the installation practice.

1. Change directories to the tutorial directory, <product\_directory>/SampleCode/tutorial. For example:

$ cd <product\_directory>/SampleCode/tutorial

1. Type the following command at the prompt:

$ gfsh start locator --name=locator1 --port=55221

The locator process runs in the background, listening for connections on port 55221.

### Start two GemFire Cache Server by using gfsh

1. From the tutorial directory, start the first cache server:

$ gfsh start server --name=**server1**   
--locators=localhost[55221]   
--cache-xml-file=./xml/server.xml --server-port=0

This command will automatically create a working directory for this server named **server1** under the <product\_directory>/SampleCode  
/tutorial directory where you executed the command.

1. Next start a second cache server by executing the following command in the same terminal:

$ gfsh start server --name=**server2**   
--locators=localhost[55221]   
--cache-xml-file=./xml/server.xml --server-port=0

This command will automatically create a working directory for this server named **server2** under the <product\_directory>/SampleCode

/tutorial directory where you executed the command.

You should now have two GemFire peers (cache servers) that are listening for client connections

### Start the Client Application

1. Open a seperate terminal windows, and start the client application.

$ java com.gemstone.gemfire.tutorial.Client

**Caution**: Make sure that you set the required PATH and CLASSPATH before executing the client.

1. Enter the following information from the client:
   * 1. Add the following persons with their friends:

> person Ethan Isabella Alice

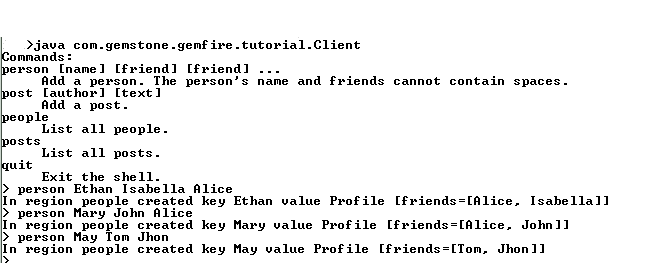
> person Mary John Alice

> person May Tom Jhon

* + 1. Notice that the following message appears for each person you add:

In region people created key Ethan value Profile [friends=[ Alice , Isabella]]

...



* + 1. Similarly, add some posts for each person. For example:

> post Ethan I like toast

> post Mary LOL!

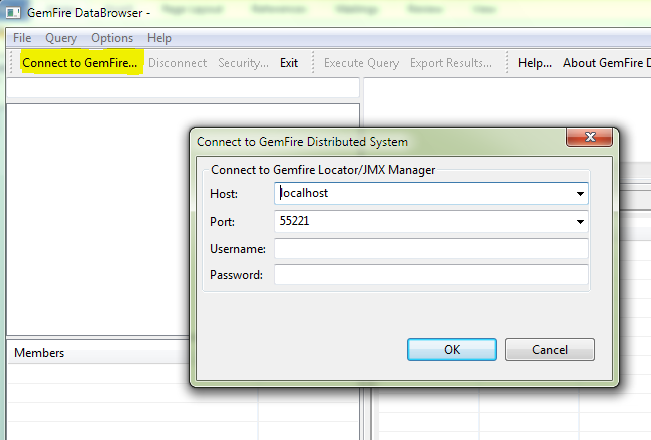
> post Ethan Hello

> post Mary Good Morning!!!

. . .

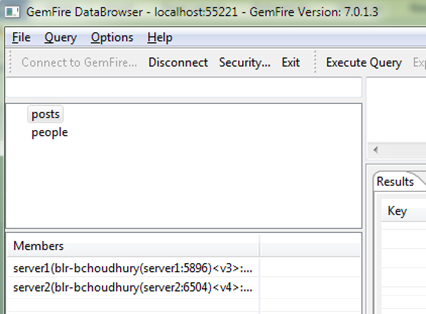
### Launch the DataBrowser Tool

1. Navigate to <product\_directory>\tools\DataBrowser\bin directory. Execute databrowser.bat to lauch the application.
2. Connect to the GemFire locator from the DataBrowser window. Provide the locator's hostname and port number:



**Note:** Alternatively, you can use gfsh to connect to the locator, and then start DataBrowser by using the 'start data-browser' command.

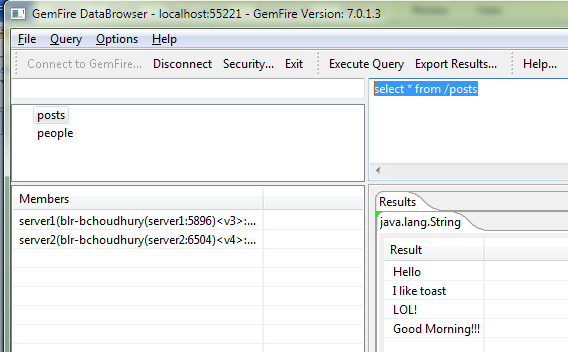
1. After connecting, you see the GemFire Regions on the left pane, and also the members hosting the Regions in the pane below.



### Execute OQL Queries

1. Enter the following query string, and click Execute Query to see the result:

Select \* from /posts



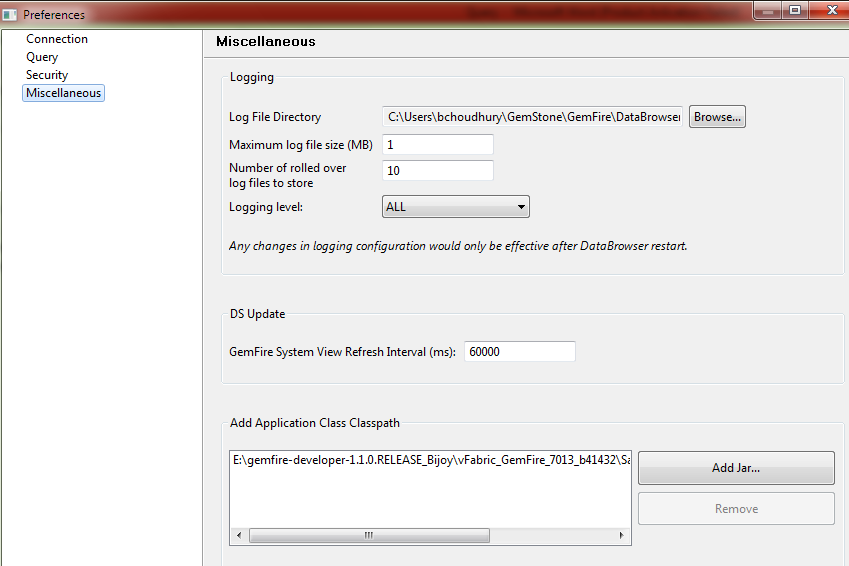
You see the results below in the Results pane.

1. By default, DataBrowser only identifies the system types. If you need to work with user-defined types, you have to import the types as JAR file within the DataBrowser. To accomplish this task, create the following JAR file:

jar cvf classes.jar com\gemstone\gemfire\tutorial\model\Profile.class   
com\gemstone\gemfire\tutorial\model\PostID.class

You can find these class files in the <product\_directory>/SampleCode/tutorial/classes directory

1. Import the JAR file within DataBrowser. Click **Options > Preferences**. In the following window, add the JAR file you created as follows:



1. Execute some more queries. For example,
2. To find all the post from Ethan:

select p.value from /posts.entries p   
WHERE p.key.author='Ethan'

1. To find all people who have listed Alice as a friend

select p.key from /people.entrySet p, p.value.friends f where f='Alice'

### Shutdown the Servers, Locator

1. Exit Client: Enter 'Quit'
2. Shutdown server1: Enter

$ gfsh -e "connect --locator=localhost[55221]" -e   
 "stop server --name=server1"

1. Shutdown server2: Enter

$ gfsh -e "connect --locator=localhost[55221]" -e   
 "stop server --name=server2"

1. Shutdown locator: Enter

$ gfsh -e "connect --locator=localhost[55221]" -e   
 "stop locator --name=locator1"