**Instruction Sheet** **Mar 24, 2005**

Setting up the example program demonstrating a Grid Service that acts as a web service client to the Amazon Web Service (AWS)

**Throughout the document, wherever you see `<username>` replace it with your actual username (mhvora, etc.)**

### Install "tutorial" files and familiarize yourself with the *CSE*Grid

You should have downloaded the "tutorial" files and installed it in your home directory in the *CSE*Grid (cerf, mills, vixie). Note that your home directory is common to all three machines. You just need to install this through one machine.

Also, request your certificate and familiarize yourself with the environment through the examples given in the tutorial. Make sure you have appropriate environment variables setup in your `.cshrc` file.

### Get Subscriber ID / Developer Token

You will need a "subsriber ID" from Amazon Web Services. This is used by your gridservice (which is a webservice client) to talk to the Amazon Web Service. It is a 20-characted code.
`http://www.amazon.com/gp/aws/registration/registration-form.html`

### Download example code

Download the example code from
`http://www.cse.buffalo.edu/~mhvora/4587/samples/amazon.tar`
OR
`/projects/Spring_2005/cse587/mhvora/public/globus/amazon.tar`
(zip versions also available at the same locations)

### Untar example code into the tutorial directory

Copy the tar file into the tutorial directory and untar the contents from there. The tutorial contains the following:

- `AmazonSearchService (package org.globus.progtutorial.services.amazon.impl)`
- Amazon web service client stubs (`package org.globus.progtutorial.services.amazon.impl.aws`)
- `AmazonSearchClient (package org.globus.progtutorial.clients.amazon)`
- `AmazonSearch.gwsdl` (located in the `schema/progtutorial/AmazonSearchService` directory)

### Move service and schema files

- Move the content of the directory:
  `$TUTORIAL_DIR/org/globus/progtutorial`
  to:
  `$TUTORIAL_DIR/org/globus/<username>_progtutorial`
- Move the content of the directory:
  `$TUTORIAL_DIR/schema/progtutorial`
  to:
  `$TUTORIAL_DIR/schema/<username>_progtutorial`

### Execute `change.sh`

From `$TUTORIAL_DIR` execute `change.sh` by typing:

```
./change.sh
```

This will change the package names and classnames in various `.java` files, in the deployment descriptor and in the `.gwsdl` file.

**Modify `namespace2package.mappings` file**
Add the following three lines to this file located in the tutorial directory:

- `http\://www.globus.org/namespaces/2005/03/amazon/AmazonSearchService=org.globus.<username>_progtutorial.stubs.AmazonSearchService`
- `http\://www.globus.org/namespaces/2005/03/amazon/AmazonSearchService/bindings=org.globus.<username>_progtutorial.stubs.AmazonSearchService.bindings`
- `http\://www.globus.org/namespaces/2005/03/amazon/AmazonSearchService/service=org.globus.<username>_progtutorial.stubs.AmazonSearchService.service`

Make sure that each of these three lines do not have any linebreaks inserted in between.

**Modify service code**
You need to insert your subscriber ID into the service code. Edit the file:
`$TUTORIAL_DIR/org/globus/<username>_progutorial/services/amazon/impl/AmazonSearchImpl.java`

At line 14, replace the value set to the variable `developerToken` to your subsriber ID.

**Build the service**
Build the service using the following command (same line):

```
./build.sh org/globus/<username>_progtutorial/services/amazon
      schema/<username>_progtutorial/AmazonSearchService/AmazonSearch.gwsdl
```

**Deploy `.gar`**
This is done by copying the generated `.gar` (which is stored in the `build/lib` directory inside the tutorial directory) to either of:

- `/home/csgrad/sjlobo/cerfgars`
- `/home/csgrad/sjlobo/millsgars`
- `/home/csgrad/sjlobo/vixiegars`

depending on which machine you are working with.

An automated shell script deploys this onto the respective instance of Globus.

**`setenv.csh`**
After you deploy any new **`.gar`** files, you must
`source $GLOBUS_LOCATION/setenv.csh`
to make sure that the newly generated `jar` files are in your classpath.

**Start the container**
Goto $GLOBUS_LOCATION. Execute
`ant startContainer -Dservice.port=`***`XXXXX`***
The XXXXX must be replaced by one of the port numbers from the port number range allocated to you.

You should see AmazonSearchService in the listing of services.

**Compile and execute client**

Compile the client by using:

```
javac org/globus/<username>_progtutorial/clients/amazon/AmazonSearchClient.java
```

Execute it by using:

```
java org/globus/<username>_progtutorial/clients/amazon/AmazonSearchClient
      http://localhost:XXXXX/ogsa/services/<username>_progtutorial/amazon/AmazonSe
      archService "author_name"
```

`author_name` represents the name of the author, for whom you would like to search for. You should see a list of book titles published by this author. These titles are fetched using the Amazon Web Service.