

# GT3@IU: User's Manual

Marcus Christie

November 6, 2003

## 1 Getting Started

### 1.1 Setting up your environment

Begin by sshing to a machine with GT3 installed. Do the following:

1. Set `JAVA_HOME` to path of a JDK 1.4.1+ installation. This varies depending on architecture and the shell you are using. On Solaris with a Bourne type shell, you would do the following:  

```
export JAVA_HOME=/1/j2sdk1.4.2
```
2. Set environment variable `ANT_HOME` to `/1/local/packages/jakarta-ant/1.5.3-1`.
3. Set environment variable `PATH` to `$JAVA_HOME/bin:$ANT_HOME/bin:$PATH`.
4. Change to directory `/usr/local/globus`.
5. Set environment variable `GLOBUS_LOCATION` to `'pwd'`.
6. Source `$GLOBUS_LOCATION/etc/globus-user-env.[c]sh`, depending on whether you are using a C shell or a Bourne shell derivative.
7. Set environment variable `LD_LIBRARY_PATH` to `/1/gcc322/lib:$LD_LIBRARY_PATH`.
8. Set environment variable `OPTIONS` to `-Dorg.globus.ogsa.client.timeout=120000`.

This environment variable sets the socket timeout for SOAP transactions to 2 minutes (120,000 ms). The default is 1 minute. If you find yourself getting "Socket Timeout Exceptions", you should first check that this option is correctly set on both the **server** and **client** ends of the transaction. If they are set correctly, then you may need to increase the socket timeout to more than 120,000 ms.

### 1.2 Get your certificate

See Section 3 for instructions on obtaining your user certificate from the NCSA CA.

### 1.3 Launch Job

To be able to launch a job against a GT3 container, you first need to run `grid-proxy-init`. This creates a proxy certificate for you and stores it as `/tmp/x509up_u[user number]`, where `[user number]` is your user number as given in `/etc/passwd`. Of course, you'll also need to have a running GT3 container to launch the job against. If one isn't available, you can use the standalone container provided with GT3. For information on how to use this container, see Section 2.

To launch a job, you use the `MasterForkManagedJobFactoryService`. GT3 comes with a test job that can be launched as a test that you have everything set up correctly. To run this test, change to the `/usr/local/globus` directory and issue the following command:

```
managed-job-globusrun -factory \  
http://host:port/ogsa/services/base/gram/MasterForkManagedJobFactoryService \  
-file etc/test.xml
```

Replace *host* and *port* with the appropriate values. If this test runs successfully, you will see:

```
WAITING FOR JOB TO FINISH  
===== Status Notification =====  
Job Status: Done  
=====  
DESTROYING SERVICE  
SERVICE DESTROYED
```

The files `stdout` and `stderr` will be created in your home directory. `stdout` should contain:

```
12 abc 34 pdscaex_instr_GrADS_grads23_28919.cfg pgwynnel was here
```

## 2 Working with the Standalone Container

The standalone container that comes with GT3 is meant to be used only for testing. This makes it excellent for debugging and polishing a Grid Service you may be working on before deploying it to a production GT3 container. The standalone container is relatively simple to work with. In this section we'll see how to start and how to stop the container.

### 2.1 Deploying/Undeploying a Grid Service to the Container

First of all, you need to generate a `gar` file for your Grid Service. Please see the [GT3 Programmer's Tutorial](#) for information and helpful scripts for generating `gar` files.

To work with the GT3 container, you'll need to `ssh` to the machine that you want to work with as user `globus`. If you haven't already done so, set up the environment for user `globus` as described in Section 1.1. Begin by changing to the `/usr/local/globus` directory. Issue the following:

```
ant deploy -Dgar.name=/path/to/gar_name.gar
```

Where `gar_name` is the name of your gar file.

To undeploy a Grid Service, change to the `/usr/local/globus` directory and run the following:

```
ant undeploy -Dgar.id=gar_name
```

For example, let's suppose you had deployed the gar file `myGridApp.gar`. To undeploy it you would use `ant undeploy -Dgar.id=myGridApp`.

## 2.2 Starting the GT3 container

To work with the GT3 container, you'll need to ssh to the machine that you want to work with as user `globus`. If you haven't already done so, set up the environment for user `globus` as described in Section 1.1. To start the container run the following:

```
globus-start-container [-p port-number]
```

The last argument is optional. If you don't specify a port number, the container will start up on port 8080. If you experience problems working with the container you started, one of the first things to try is to start it on a port other than 8080<sup>1</sup>.

When started, the container should print to standard output a list of url's pointing to all of the grid services that it is now running. If you've just deployed a new grid services, you'll want to check this list and make sure that your service is listed. If not, you will probably need to redeploy your service.

## 2.3 Stopping the GT3 container

To stop the GT3 standalone container (assuming you are user `globus` and have set up your environment correctly), issue the following:

```
globus-stop-container -secure [soft http://localhost:port]
```

This assumes you are stopping the container on the same host as the one that is running the container. You only need the optional last argument if you started the container on a port other than 8080. If you did, replace `port` with the port you started the container on.

# 3 Managing User Certificates

Our user certificates are provided by the NCSA CA. See the following links for more information.

[Getting your certificate](#)

[User your NCSA certificate](#)

Essentially, you'll need to follow the first link and follow its instructions in order to generate a grid certificate. Then you'll need to copy it to your `~/globus` directory on your local account to be able to use it locally. Then you'll be able to issue a `grid-proxy-init` and such with the NCSA certificate.

---

<sup>1</sup>On some machines, port 8080 is firewalled.

## 4 Troubleshooting

### 4.1 Cleaning up UHE's

### 4.2 Linux machine doesn't recognize "localhost"