

GT 4.2.0 CoG jGlobus: System Administrator's Guide

DRAFT

GT 4.2.0 CoG jGlobus: System Administrator's Guide

Introduction

This guide contains installation and configuration information for system administrators installing CoG jGlobus (known outside of GT as the Java CoG Kit). It explains how to install, configure and test the installation.

Important

This information is in addition to the basic Globus Toolkit prerequisite, overview, installation, security configuration instructions in the [Installing GT 4.2.0](#). Read through this guide before continuing!

DRAFT

Table of Contents

- 1. Building and installing 1
 - 1. Building from source 1
 - 2. Installing binary distribution 1
- 2. Configuring 3
 - 1. Trusted Certificates Location 3
 - 2. Signing Policy Location 3
 - 3. User Certificate Location 3
 - 4. User Private Key Location 4
 - 5. Proxy file Location 4
 - 6. Public IP address 4
 - 7. TCP Port Range 4
- 3. Deploying 5
- 4. Testing 6
- 5. Security considerations 7
 - 1. Security considerations for CoG jGlobus 7
- 6. Troubleshooting 8

DRAFT

Chapter 1. Building and installing

Java CoG Kit is distributed as part of a default GT 4.2.0 installation. For basic installation instructions, see the [Installing GT 4.2.0](#). No extra installation steps are required for this component.

The following are optional instructions for more advanced types of installations. These are for those advanced users who want to build the latest code from CVS or are just interested in the Java CoG Kit.

1. Building from source

1.	<p>Obtain the source code for Java CoG Kit:</p> <p>From CVS.</p> <ol style="list-style-type: none"> To get the latest source from CVS execute: <pre>cvs -d :pserver:anonymous@cvs.globus.org:/home/dsl/cog/ CVS \ checkout jglobus</pre> Change into the jglobus directory. <pre>cd jglobus</pre> <p>From source distribution.</p> <ol style="list-style-type: none"> Untar or unzip the distribution archive. <pre>tar xvfz cog-XXX-src.tar.gz</pre> Change into the unpacked distribution directory. <pre>cd cog-XXX</pre>
2.	<p>Run:</p> <pre>ant all</pre>

2. Installing binary distribution

1.	<p>Untar or unzip the distribution archive.</p> <pre>tar xvfz cog-XXX-tar.gz</pre>
2.	<p>Change into the unpacked distribution directory.</p> <pre>cd cog-XXX</pre>

3.	<p>Set the COG_INSTALL_PATH environment variable to the unpacked distribution directory.</p> <p>On Windows:</p> <pre>set COG_INSTALL_PATH=c:\cog-1.2</pre> <p>On Unix/Linux:</p> <pre>setenv COG_INSTALL_PATH /soft/cog-1.2/</pre> <p>or</p> <pre>export COG_INSTALL_PATH=/soft/cog-1.2/</pre>
----	--

DRAFT

Chapter 2. Configuring

In general no extra configuration is required to use GSI Java. However, in certain cases additional configuration might be required. The GSI Java configuration file (based on Java CoG Kit) is placed in `$HOME/.globus/cog.properties` on Unix/Linux machines or `%USERPROFILE%\globus\cog.properties` on Windows machines.

1. Trusted Certificates Location

Trusted certificates can be configured as described below and are loaded in the specified order:

1. Property `cacerts` in configuration file `cog.properties` file can be used to set the trusted certificates. The value of the property is a list of comma-separated CA certificates or directories of certificates. Example:

```
cacert = /home/gawor/MyCaCert.pem, /etc/grid-security/certificates/
```

If a directory is specified in the list, all certificate files within that directory with `.<digit>` extension will be loaded.

2. The system property `X509_CERT_DIR` can be used to configured trusted certificates. It is used if the `cog.properties` file is not found or the `cacert` property is not set
3. Trusted certificates can be placed in `${user.home}/.globus/certificates` directory. If system property is not set, this location is checked.
4. The certificates can be placed in `/etc/grid-security/certificates` directory. This directory is checked on Unix/Linux machines if certificates are not found as described in previous steps.

If one of the above directories with certificates is found, all the certificates in that directory will be loaded and used. If no directory is found, GSI Java will not work.

2. Signing Policy Location

Signing policies are read in from the same directory as trusted certificates. All files with `<caHash>.signing_policy` in the same directory as trusted certificates are loaded for processing. By default signing policies are required and enforced.

Signing policy enforcement can be disabled by setting a system property `java.security.gsi.signing.policy` to "no" or "false".

Details on signing policy grammar support is outlined [here](http://dev.globus.org/wiki/Signing_Policy)¹

3. User Certificate Location

User certificate can be configured as described below and an attempt is made to find the file in the order specified:

1. Location of the file can be specified as value of property `usercert` in configuration file `cog.properties`
2. Location of the user certificate can be set as value of environment variable `X509_USER_CERT`. This property is used if `cog.properties` file is not configured or the `usercert` property is not specified in the file.

¹ http://dev.globus.org/wiki/Signing_Policy

3. The certificate can be placed in default location `${user.home}/.globus/usercert.pem`. If none of the above configuration is found, the library defaults to this location.

4. User Private Key Location

User private key can be configured as described below and an attempt is made to find the file in the order specified:

1. Location of the file can be specified as value of property `userkey` in configuration file `cog.properties`
2. Location of the user private key can be set as value of environment variable `X509_USER_KEY`. This property is used if `cog.properties` file is not configured or the `userkey` property is not specified in the file.
3. The private key can be placed in default location `${user.home}/.globus/userkey.pem`. If none of the above configuration is found, the library defaults to this location.

5. Proxy file Location

Proxy file can be configured as described below and an attempt is made to find the file in the order specified:

1. Location of the file can be specified as value of property `proxy` in configuration file `cog.properties`
2. Location of the proxy can be set as value of environment variable `X509_USER_PROXY`. This property is used if `cog.properties` file is not configured or the `proxy` property is not specified in the file.
3. If the above configuration is not used, then the library looks for proxy in default location using the following rules:
 - If a UID system property is set, and running on a Unix/Linux machine it returns `/tmp/x509up_u${UID}`.
 - If on any other machine than Unix/Linux, it returns `${tempdir}/x509up_u${UID}`, where `tempdir` is a platform-specific temporary directory as indicated by the `java.io.tmpdir` system property.
 - If a UID system property is not set, the username will be used instead of the UID. That is, it returns `${tempdir}/x509up_u${username}`

6. Public IP address

Sometimes, especially on machines with DHCP enabled, the local IP address of the machine might be incorrectly detected by Java or the wrong non-public IP address might be used if behind a NAT router or on VPN. To configure the public IP address used by GSI Java, add the following line to the configuration file:

```
ip=<current ip>
```

This has the same effect as setting the `GLOBUS_HOSTNAME` environment property or Java system property.

7. TCP Port Range

Sometimes it is necessary to restrict the listening port numbers that Java CoG Toolkit will use. To set the port range in the configuration file, just add the following line:

```
tcp.port.range=<min>, <max>
```

This has the same effect as setting the `GLOBUS_TCP_PORT_RANGE` environment property or Java system property.

Chapter 3. Deploying

This section does not apply to the Java CoG Kit.

DRAFT

Chapter 4. Testing

There are no tests to test the Java CoG installation. However, the unit tests are available with the source distribution. Since some of the unit tests rely on GT2 services to be running, first make sure that they are running ok. Also, you will need to configure the tests (look for various `test.properties` files) in the source distribution with the appropriate information (for example the right service port, hostname, etc.). Finally, to run the tests do:

```
ant test
```

DRAFT

Chapter 5. Security considerations

1. Security considerations for CoG jGlobus

1.1. Functions that execute an external program

Under some circumstances, the `org.globus.util.Util.setFilePermissions()` and the `org.globus.util.ConfigUtil.getUID()` functions execute an external program; thus, its behavior is influenced by environment variables such as the caller's `PATH` and the environment variables that control dynamic loading. Care should be used if calling these functions from a program that will be run as a Unix `setuid` program, or in any other manner in which the owner of the Unix process does not completely control its runtime environment.

1.2. Permissions of proxy files

Since Java does not provide an API for setting the permissions of a file, the Java CoG Kit will attempt to execute the `/bin/chmod` program in the background to set the permissions of the given file. If that program cannot be executed for any reason or fails to execute correctly, a proxy file might end up with incorrect file permissions (depending on `umask` setting). Usually a warning will be displayed if that occurs (especially on Windows since `/bin/chmod` is not supported on that platform).

Chapter 6. Troubleshooting

Please see the [FAQ](#)¹ and the [web page](#)² for any problems concerning Java CoG Kit. Also, you can send email describing your problem to the java@globus.org³ mailing list ([must subscribe first](#)⁴) or search for the problem in the [archives](#)⁵.

DRAFT

¹ <http://www.globus.org/cog/distribution/1.2/FAQ.TXT>

² <http://www.globus.org/cog/java/>

³ <mailto:java@globus.org>

⁴ <http://www.globus.org/subscriptions.php>

⁵ http://www.globus.org/mail_archive/java/threads.html