



The Grid Job Handler of the Fraunhofer Resource Grid

Fraunhofer FIRST, Berlin

Version: \$Name: ver20040308\$
\$ID\$

1 Introduction

This is a short instruction for the program package de-fhrg-jobhandler, version 20040308, to submit grid jobs on the Fraunhofer Resource Grid (FhRG). In addition to the execution of simple, uncoupled applications which have been defined by the Globus-specific description language XML-RSL, the current version of the Grid Job Handler for the first time allows the execution of coupled grid jobs based on the description language GJobDL, version 1.1. Here, the control and data flow is described by a Petri net. The Grid Job Handler uses the Globus service GRAM in order to execute the single components of the coupled grid application on usual Globus grid nodes. Utilizing the service GSI-FTP you can transmit the necessary data and executable files to the target computer before the job execution. The installation of Globus on the local machine on which the Grid Job Handler will be started is not required.

2 Installation

2.1 System Requirements

To run the programs a Java Virtual Machine (JVM), version 1.4 or higher, is required. To execute the Unix shell scripts in the `de-fhrg-jobhandler-20040308/bin/` directory you must set the environment variable `FHRG_HOME` to the absolute path of the `de-fhrg-jobhandler-20040308` directory. Furthermore, a shell compatible with the bash should be installed in `/bin/bash`. To compile the sources using the `build.xml` file, `ant` version 1.3 or higher is required. The programs have been tested with Linux (Debian GNU/Linux, kernel 2.4.19), and MS-Windows XP with Java version 1.4.1.

The installation of Globus on the local machine is not required. The Grid Job Handler accesses Globus using the CVS version of the Java CoG kit of July 23rd, 2003 [4], which is part of the program package, but in a slightly modified version. On the grid computers used for the execution of the grid applications, Globus version 2.4 and GSI-FTP must be installed at the usual ports. In case of existing firewalls, the corresponding ports of the own as well as all other grid computers must be unblocked. You will find a list of the usual Globus services with the according port ranges at <http://www.globus.org/security/v1.1/firewalls.html>. [5]

Furthermore, the user requires a valid user certificate and the corresponding private and public keys which have been certified by a Certificate Authority (CA) accepted in the grid. In addition, the public certificates of the CAs which are trusted by the user are required.

2.2 Configuration

At first, the user certificate (X509) for Globus and the corresponding private key must exist on the local machine. Please note that the private key must not be shown to other persons than the owner. Thus, reading permissions must be set accordingly. The private key must not be transferred to other computers via insecure transfer paths, e. g. FTP or rcp. In addition, the certificate of the CA (Certificate Authority) that signed the user certificate is required. By default, the user certificates are located in the directory
~/ .globus (Linux),
C:\Windows\.globus (MS-Windows98) or
C:\Documents and Settings\\.globus (MS-Windows2000, MS-WindowsXP). For further information about Globus certificates refer to <http://www-fp.globus.org/security/v1.1/certs.html>. Users working at the Fraunhofer Gesellschaft can order certificates from the Fraunhofer CA at <http://pki.fraunhofer.de/>. Additionally, an own CA for FhRG was set up. Please contact ca@fhrg.fhg.de > for further information.

The CoG kit is configured by clicking on the menu entry **File - Configure** in the graphical user interface. The settings are saved in the `~/.globus/cog.properties` file and will not be effective unless you restart the program. You can also change the settings in the `cog.properties` file manually. In this case, the following entries are possible:

usercert: Location of the Globus user certificate (required)

userkey: Location of the private key for the user certificate. Only the owner of the private key is allowed to have reading permission for this file (required)

proxy: Location of the user security proxy. The name of the proxy is generally a combination of the string "x509up_u" and the system-dependent user ID. Only the creator of the proxy is allowed to have reading and writing permission for this file (required)

cacert: Contains a comma-separated list of the CA certificates trusted by the user (required)

org.globus.dev.random: **true** or **false** depending on the question if the Java CoG Kit should use `/dev/urandom` like Unix for the generation of random figures (optional)

random.provider: Indicates a Java provider installed in the system to be used by default for random figures (optional)

random.algorithm: Indicates the algorithm to be used for the generation of random figures by the provider (optional)

proxy.strength: Default size of the Grid proxy in bits (optional)

proxy.lifetime: Default validity of the Grid proxy in hours (optional)

ip: The IP number under which the computer can be reached from a remote location. This property may be necessary if the computer is also part of a local subnet and owns more than one IP number (optional)

http.proxyHost: The IP address of the HTTP proxy for the HTTP communication to remote locations if, e. g., the computer is secured by a firewall (optional)

http.proxyPort: The port of the above HTTP proxy (optional)

ftp.proxyHost: The IP address of the FTP proxy for the ftp communication to remote locations if, e. g., the computer is secured by a firewall (optional)

ftp.proxyPort: The port of the above FTP proxy (optional)

socksProxyHost: The IP address of the Socket proxy for the Socket communication to remote locations if, e. g., the computer is secured by a firewall (optional)

socksProxyPort: The port of the above Socket proxy (optional)

tcp.port.range: The free port range in the form of **<minport>**, **<maxport>**, to which the local net services should be constricted (optional)

usermapping: If the Grid Job Handler is run as a web service, user IDs are generally mirrored into pseudonymous certificates. With the parameter **usermapping** you can indicate the file name where to define the mirror. The **usermapping** file contains one entry per line in the form of **<user id>=<certDir>** where **<certDir>** is the directory in which the certificate (usercert.pem) and the unencrypted private key (userkey.pem) are stored (optional).

anonymous.certdir: The certificate directory for the user ID that is not included in the **usermapping** file.

Example settings in the cog.properties file in Linux or Unix

```
#Java CoG Kit Configuration File
#Tue Feb 25 22:30:30 CST 2003
usercert=/home/albert/.globus/usercert.pem
userkey=/home/albert/.globus/userkey.pem
proxy=/tmp/x509up_u1000
cacert=/usr/local/globus/share/certificates/42864e48.0
ip=140.221.56.12
tcp.port.range=6000,6060
```

Example settings in the cog.properties file in MS-Windows

```
#Java CoG Kit Configuration File
#Tue Jul 15 10:00:33 CEST 2003
usercert=C:\\Documents and Settings\\albert\\.globus\\usercert.pem
userkey=C:\\Documents and Settings\\albert\\.globus\\userkey.pem
proxy=C:\\Documents and Settings\\albert\\.globus\\x509up_u_albert
cacert=C:\\Documents and Settings\\albert\\.globus\\42864e48.0
ip=140.221.56.12
tcp.port.range=6000,6060
```

To be able to submit a grid job, you first have to create a new security proxy. To do this, click on **Security - Grid Proxy - Create**.

2.3 Directory Content

/bin: Scripts to start the program in Linux/MS-Windows

/doc: Documentation of the Grid Job Handler

/examples: Examples for the RSL-XML and the GJobDL descriptions of grid applications. To execute these grid applications Globus access privileges on the according grid nodes are required.

/jar: The used Java classes and other files in the form of jar packages

/javadoc: The HTML documentation for the Java packages de.fhrg.** and the Java CoG kit (packages org.globus.**)

/log: Log files

/resources: Files that are required independently of the source code to execute the programs. These files are additional parts of the jar packages.

/sources: The sources of the Java packages de.fhrg.** and of the used and slightly modified Java CoG kit. A description of the Java packages can be found in the HTML documentation under /javadoc.

3 Features

3.1 Usage

The Grid Job Handler can be started in several different modes: Using the class **de.fhrg.jobhandler.JobHandler**, you can operate the Grid Job Handler via command line parameters, using the class **de.fhrg.jobhandler.desktop.JobHandlerDesktop** you get a graphical user interface. The Grid Job Handler can also be provided as a web service.

3.1.1 Graphical User Interface

To start the graphical user interface you can use the scripts **de-fhrg-jobhandler-desktop.sh** (Linux) or **de-fhrg-jobhandler-desktop.bat** (MS-Windows) in the `de-fhrg-jobhandler-20040308/bin` directory. You can also choose to enter the command

```
java -cp de-fhrg-jobhandler-20040308.jar
de.fhrg.jobhandler.desktop.JobHandlerDesktop
```

directly in the command line of the `de-fhrg-jobhandler-20040308/jar` directory to start the graphical user interface.

3.1.2 Command Line Tool

As a command line tool in Linux, the Grid Job Handler can be started using the script **de-fhrg-jobhandler.sh** or directly using the command

```
java -cp de-fhrg-jobhandler-20040308.jar
de.fhrg.jobhandler.JobHandler. In this case, the following command
line parameters are valid:
```

```
de.fhrg.jobhandler.JobHandler
  -filename <XML_FILENAME>
  [-jobType <JOB_TYPE>]
  [-v <VERBOSE_LEVEL>]
  [-home <FHRG_HOME>]
  [-schedulerType <SCHEDULER_TYPE>]
```

<XML_FILENAME>: File name of the XML file which contains the description of the grid applications.

<JOB_TYPE>: Either "GJobDL" or "RSL". The default value is "GJobDL" (optional)

<VERBOSE_LEVEL>: Amount of log output (optional): 0: only error messages up to 5: all debug messages

<FHRG_HOME>: The installation directory of the Grid Job Handler (e. g. /home/albert/de-fhrg-jobhandler-20040308). FHRG_HOME can also be set as environment variable.

<SCHEDULER_TYPE>: Type of the scheduler to be used (optional). Either "test" or "MDS". From the list of possible resources, the test scheduler always selects the hardware resource that has not been used for the longest time. The MDS scheduler, though, checks the LDAP server of the FhRG and selects the fastest computer considering the current workload. The MDS scheduler is still a test version that does not work faultless yet.

3.1.3 Grid Job Handler Web Service

The Grid Job Handler can also be accessed by a web service. In the package **de.fhrg.jobhandler.webservice** you will find the necessary classes. An example instruction for the deployment of the Grid Job Handler with axis and tomcat can be found at **doc/jobHandlerWebServiceDeployment.txt**. Currently, the users are assigned to pseudonymous certificates based on a user assignment file. For further information see chapter 0.

The Grid Job Handler web service can be accessed by different ways. An example is the command line client in the package **de.fhrg.jobhandler.webservice.client**. To start this command line client you can use the script **de-fhrg-jobhandler-webservice-client.sh**. The command line parameters are described in analogy to the ones in chapter 0. The applets in the package **de.fhrg.jobhandler.applet** also access the Grid Job Handler Web Service.

3.2 Description of Grid Applications with the GJobDL

The concept and specifications of the GJobDL are described in detail in 5.2, [1] and [3]. Here, we only describe the implementation-specific properties which cannot be found in the above mentioned sources. Some helpful examples of GJobDL descriptions of Grid applications can be found in the **<FHRG_HOME>/examples** directory. The described examples are still quite extensive as the current version of the Grid Job Handler only supports automatic refinement of the Petri nets for partial areas. The GJobDL documents can be created either manually with an appropriate XML editor or using the **Grid Job Builder** which is being developed by Fraunhofer IGD.

3.2.1 Method Calls in Transitions

Transitions in GJobDL documents may contain method calls or conditions. The following is an example of the definition of a method call:

```
<transition id="singleTransfer">
  <methodCall>transferData ()</methodCall>
</transition>
```

In the current version, the Grid Job Handler supports the following method calls (see also the interface `de.fhrg.jobhandler.MethodCalls`):

transferExecutable(String softwareTransitionId): Transfers the executable file or the archive of a software component to the place of execution. If the place of execution is not yet determined it will be set by means of resource mapping and scheduling.

unpack (): Unpacks all tar.gz files which are linked to the input places of this transition.

transferData(String softwareTransitionId): Transfers the files, which are linked to the input places of this transition, to the place of execution of the software component of the given software transition. If the place of execution is not yet determined it will be set by means of resource mapping and scheduling.

transferData (): Transfers the file, which is linked to the input place of this transition, to all places defined by the output places.

clear (): Clears the used computing nodes by deleting all temporary directories created with the grid job.

3.2.2 Conditional Transitions

Furthermore, you can define conditional transitions, e.g.:

```
<transition id="count.lt.7">
  <condition>countLT(7)</condition>
</transition>
```

At the moment, the Grid Job Handler supports the following conditions (see also the interface `de.fhrg.jobhandler.Conditions`):

isDone (): This condition is fulfilled when all input places contain a Boolean with the value "true".

isFailed (): This condition is fulfilled when at least one input place contains a Boolean with the value "false".

countLT(int maxCount): Counts the number of times which a transition has fired. If the number is smaller than maxCount, the condition is fulfilled and the transition fires.

negate(String transitionID): With this condition you can negate the condition of another transition (corresponds to ELSE when using IF THEN), e.g.

negate(count.lt.7) only fires when the condition of the transition "count.lt.7" has not been fulfilled.

timeGE(long msSince1970): Compares the current system time with the given timestamp. The transition only fires if the current system time is higher or equal to the given time. The time format corresponds to the Java time format used in the date class (milliseconds since 1.1.1970 00:00 UTC).

idleTimeGE(long milliseconds): Compares the duration after the transition was last fired with the given duration. The transition only fires if after the last firing the given time in milliseconds has passed.

3.3 Hot Keys

<Ctrl>+<q>	Exits the program
<Ctrl>+<t>	Scheduler "Test" is used
<Ctrl>+<m>	Scheduler "MDS" is used
<Ctrl>+<l>	Loads GJobDL document from local file
<Ctrl>+<r>	Loads GJobDL document from repository
<Ctrl>+<s>	Saves current GJobDL document in local file

3.4 Changes

Version 20040308 – Date: 2004-03-08

Changes from version 20031127

- Java VM 1.4.0 or higher is required because of JavaCoG.
- New method call "clear()" deletes all created temporary directories of a grid job.
- Mouse cursor in the shape of an hourglass when the desktop handler is occupied.
- Automatic line wrap when displaying error messages.
- Connection to PBS and other Globus job managers. You can indicate a type (single, multiple, mpi, condor) in the software resource description using <parameter name="jobType"> .
- Indication of geographical coordinates of hardware resources in the event channel according to ISO6709 for the presentation on maps.
- New version of the scheduler. You can now choose between a standard scheduler (formerly "test"), an internal MDS, and an external Client/Server MDS scheduler.
- Automatic refinement of the Petri net for software installation (software deployment).
- Arrangement of Petri nets in a reproducible manner.
- New color scheme for the display of Petri nets.

Version 20031127 – Date: 2003-11-27

Changes from version 20030708

- Documentation now in /doc.
- Grid Job Handler Client accesses the Grid Job Handler via a web service.
- Grid Job Handler exists as a Web Service now.
- Grid Job Handler supports pseudonymous certificates and thus creates X509 proxies with limited durability.
- New Axis libraries (version 1.1).
- Additional events for the dynamic refinement of the Petri net: addTransition, addPlace, addArc, removeArc.
- Update CoG1.1a to the CVS version of July 23rd, 2003.
- New conditions for transitions: timeGE and idleTimeGE.
- Implementation of Touchgraph to display the Petri nets. Therefore, the arrangement of the graphs by an external C program is unnecessary.
- Progress display when loading the resource descriptions and creating the Grid jobs.
- New Interface to the Scheduler.
- New: Security → Grid Proxy → Information: Indicates the name and the remaining validity of the currently used security proxy.
- Grid job status: undefined, defining, pending, active, suspended, failed, done. The status is transferred by the Java beans propertyChange-Mechanismus.

Version 20030708 – Date: 2003-07-08

Changes from version 20030424:

- Update CoG-0.9.13 → CoG1.1a with a new encryption library (Bouncycastle) with better license conditions.
- Also works with Globus 2.4.
- Uses GADL1.1: Schema instead of DTD, resource identifier corresponds to the URL where the resource description is available.
- The Grid Job Handler accesses the distributed resource descriptions via the internet if they are not defined in the local GJobDL document.
- Default directory is now FHRG_HOME+"/examples".
- Shell scripts now use bash instead of tcsh.
- Alpha version of a map of Germany with geographical representation of the grid jobs.
- The configuration of the local IP number and the internet proxy using the graphical user interface is not available at the moment and will be re-integrated in a later version.
- Many smaller bug fixes.

Version 20030424 - Date: 2003-04-24

Changes from version 20021113:

- Automatic refinement of the Petri nets for data transfer.
- Built-in first version of the MDS Scheduler by ITWM.
- In the configuration wizard, you can now indicate the local IP number to be used for external communication.
- This is necessary for computers with two IP numbers (internal/external network). First connection to the repository with SOAP.
- Bug fix for problems with <executionLocation> in the GJobDL.

- GEM for automatic arrangement of the Petri nets now for MS-Windows available.
- The JobHandlerDesktop can be used to change and save the GJobDL descriptions.
- PetriNetFactory and GridResourceFactory for the creation of instances to represent Petri nets and Grid resources.

Version 20021113 - Date: 2002-11-13

Changes from version 20021002:

- New command line parameter: -home.
- MethodCalls and Conditions added. Allowed MethodCalls: transferExecutable(String), transferData(String), transferData(), unpack(). Allowed Conditions: isDone(), isFailed(), countLT(int), negate(string).
- Graphical presentation of the process flow in a Petri net (monitoring).
- Transitions and places have a status (IDLE, ENABLED, FIRING, ...).
- When submitting software components, the directory will be set: RSL: (directory = \$(HOME) # "/...")
- Enhanced user interface on the JobHandler desktop.
- Some new examples of GJobDL descriptions added.
- Bug fix: Transition fires only when ALL corresponding atomicJobs are finished (=DONE or FAILED).
- Bug fix: Temporary file locations are treated as such.
- X/Y coordinates of places and transitions are taken into consideration.
- Globus proxy is delivered separately with every AtomicJob.

Version 20021002 - Date: 2002-10-02

Changes from version 20020214:

- GJobDLExecutor was renamed to JobHandler.
- Changed package structure (de.fhrg.first.** -> de.fhrg.jobhandler.*, de.fhrg.util.*, ...).
- Encapsulation of GRAM jobs with the abstract class AtomicJob.
- AtomicGramJob now extends the class AtomicJob.
- New type of AtomicJob: AtomicGridFtpJob. Uses the
- Package de.fhrg.gridftp for the transmission of files by GSI-FTP.
- New command line parameters: -jobType, -home, -v, -filename.
- GADL 0.9 is basically supported.
- The GJobDL document can be read using a validating DOM parser.
- Petri nets with transitions, arcs and places.
- Resource Mapping: Resolves dependencies between grid resources.
- Scheduler: Simple scheduling without using additional information about grid resources.
- Update cog-0.9.12 -> cog-0.9.13.
- New Version of iaik_*.jar.

Version 20020214 - Date: 2002-02-14

Changes from version 20020129:

- Monitoring of grid jobs.
- Changing the XML parser to Xerces 2.0.0.

- Supporting the proxies when accessing the internet (is especially required by computers behind a firewall). Configuration of the proxies using the configuration wizard. You start the wizard by clicking the menu item File>Configure on the graphical user interface.
- Improved error management.

Version 20020129 - Date: 2002-01-29

First version for submitting grid jobs which were defined with the XML-RSL description. Uses CoG kit version 0.9.12. This version does not have any functionalities exceeding the normal Globus implementation. Essential functions of this version:

- Environment configuration (keys, certificates, ...) using a configuration wizard.
- Creating and reviewing a grid proxy.
- Loading grid job descriptions in XML-RSL format.
- Submitting the grid jobs using the GRAM service of Globus.
- Displaying this help file.

3.5 Bugs, Suggestions, Contributions

Please contact Andreas Hoheisel via e-mail
<andreas.hoheisel@first.fraunhofer.de>

Known bugs:

- Exit status of the application is not transferred correctly to the Grid Job Handler by GRAM. The Grid Job Handler receives the status "done" even if the exit status of the executed programs is not 0. This is not a bug of the Grid Job Handler, but a bug/feature of Globus.
- Not all of the possibilities given by the GJobDL are implemented in the Grid Job Handler, e. g. the comparison using operators as well as the authentication.
- Partly, the URL of the Grid Job Handler web service is hard-coded, but should be configured with the configuration tool.

4 Outlook

For the next versions of the program package de-fhrg-jobhandler we have planned the following extensions:

- Support of additional "Conditions" and "MethodCalls" of transitions in GJobDL applications.
- Additional authorization using an access control component (SIT).
- Complete connection to the FhRG repository (IAO).

5 Further Information

5.1 Web Sites

- Fraunhofer Resource Grid: <http://www.fhrg.fraunhofer.de>
- Fraunhofer FIRST: <http://www.first.fraunhofer.de>
- Fraunhofer Gesellschaft: <http://www.fraunhofer.de>
- Fraunhofer CA: <http://pki.fraunhofer.de>
- OpenSource eXeGrid: <http://www.exeGRID.net>
- Java CoG Kit: <http://www.globus.org/cog/java>
- RSL: http://www-fp.globus.org/gram/rsl_spec1.html
- Touchgraph: <http://www.touchgraph.com>
- Andreas Hoheisel: <http://www.andreas-hoheisel.de>

5.2 Literature

- [1] Hoheisel, Andreas, Uwe Der. An XML-Based Framework for Loosely Coupled Applications on Grid Environments. In P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 245-254, 2003. <http://www.andreas-hoheisel.de>
- [2] Hoheisel, Andreas, Uwe Der. Dynamic Workflows for Grid Applications. In Proceedings of the Cracow Grid Workshop '03, 2003. <http://www.andreas-hoheisel.de>
- [3] Hoheisel, Andreas. Ein Komponentenmodell für Softwarekomponenten des Fraunhofer Resource Grid, Internal Report, Fraunhofer FIRST, 2002. <http://www.andreas-hoheisel.de>
- [4] Hoheisel, Andreas. Grid Application Definition Language - GADL 0.2, Internal Report, Fraunhofer FIRST, 2002. <http://www.andreas-hoheisel.de>
- [5] von Laszewski, Gregor, Ian Foster, Jarek Gawor, Peter Lane. A Java Commodity Grid Kit, Concurrency and Computation: Practice and Experience, pages 643-662, Volume 13, Issue 8-9, 2001. <http://www.globus.org/cog/java/>
- [6] von Laszewski, Gregor, et al. The Java CoG Kit User Manual, Draft Version 1.1a, MCS Technical Memorandum, ANL/MCS-TM-259, March 14, 2003. <http://www.globus.org/cog/manual-user.pdf>

5.3 Contact

- Andreas Hoheisel <andreas.hoheisel@first.fraunhofer.de>
- Uwe Der <uwe.der@first.fraunhofer.de>
- Dr. F.-J. Pfreundt (project coordinator) <pfreundt@itwm.fraunhofer.de>
- CA (Certificate Authority) of FhRG <ca@fhrg.fhg.de>

6 Legal statement

6.1 License Conditions

We plan to release the Grid Job Handler as an open source project under the GPL. Up to this release, the copyright is with © 2003 Fraunhofer FIRST. Until further notice it is allowed to use unchanged copies of the software including this license condition for non-commercial purposes free of charge.

6.2 Trademarks and copyright

The used names and product names are possibly registered trademarks or trademarks of the corresponding companies.

This product includes software developed by and/or derived from the Globus Project (<http://www.globus.org/>) to which the U.S. Government retains certain rights.

This product includes software developed by TouchGraph LLC (<http://www.touchgraph.com/>).

6.3 Disclaimer

Fraunhofer FIRST is not liable for damages including follow-up damages, especially not for any damages on the user's hardware and software that the user has suffered from the software distributed by Fraunhofer FIRST, regardless if the damage is caused by software which is produced, licensed or used in any other way by the company.