Campus Grid Toolkit 1.1.3

Documentation

Documentation v1.1.3

Table of Contents

1 Introduction

Many institutions provide different computational and data resources for use by their researchers. However, accessing and making the most of these resources can be difficult. Campus Grid Toolkit is an easy-to-install and easy-to-use software package that makes it straightforward to access and control all of the computational and data resources available to you. By simplifying access to resources and providing a single user interface, Campus Grid Toolkit allows you to spend more time on your research so that you can explore more avenues of inquiry and run more scenarios. Ultimately, this means that you get your results earlier and publish sooner.

The Campus Grid Toolkit Server supports Linux operating systems and the Campus Grid Toolkit Client supports Windows (Windows XP only) and UNIX (i.e. Linux and Mac) operating systems. Where different commands are required for UNIX and Windows systems, the commands will be labelled `[UNIX]` or `[Windows]`.

The latest news, developments and known issues can be found on the Campus Grid Toolkit pages: http://www.omii.ac.uk/wiki/CGT and http://www.omii.ac.uk/wiki/KnownIssuesCGT.

2 Connecting Different Resources

There are many different computational and data storage resources available to researchers. Typically, each resource has its own specific language/interface that must be learnt and understood before the resource can be used. Needless to say, this requires time that would be better invested in your research. Campus Grid Toolkit overcomes this problem by providing the researcher with a single interface through which a number of different types of resource can be accessed.

The current version of Campus Grid Toolkit is provided with an automatic set up for local execution (fork), Condor and PBS. Future releases of the software will be provided with an automatic set up for further job submission systems, as described in section 12.

3 Necessary Prior Computing Experience

Campus Grid Toolkit has been designed for novice users. Historically, e-Research software has been the preserve of computational scientists. However, this precedent is starting to change as recent trends have seen the uptake of e-Research by researchers from many different disciplines. Campus Grid Toolkit's easy-to-use design aims to increase this trend by making it easy for researchers to benefit from e-Research software, regardless of their computing expertise.

This documentation has also been designed for the novice user, so programming terms are described in detail and each step in the installation process is documented. Consequently, only a basic understanding of UNIX commands will be necessary to install and set up Campus Grid Toolkit. Experienced users may find this treatment laborious, but we think that it is a necessary sacrifice to ensure that e-Research is made available to a wider audience. Experienced users may prefer the quickstart section (section 7), which covers only the main installation steps.

Targeting the documentation at novice users also means that we have chosen standard 'safe' options in the following installation instructions. To some, this may make Campus Grid Toolkit appear inflexible. It is not! Experienced users will be able to adapt Campus Grid Toolkit, as described in section 11. Furthermore, experienced users can find more information on how to adapt Campus Grid Toolkit by reading the OMII-UK Development Kit User Guide (www.omii.ac.uk/dissemination) or by contacting OMII-UK support (support@omii.ac.uk).

4 Operation of Campus Grid Toolkit

Campus Grid Toolkit allows you to submit jobs to resources so that you can take advantage of the processing power or data storage provided by those resources. A job may be a set of instructions that tell a resource to run a program on some of your data to produce other data that is useful to you. Alternatively, a job may be simply a set of instructions telling your resource to store data that you wish to preserve for later research.

Your job is sent from your client machine to a Web Service that controls how the job will be submitted to your resources. The Web Service submits the job and then passes the results, if there are any, back to your client machine or to another machine, as necessary. Two interfaces are provided with Campus Grid Toolkit: Direct GridSAM and Managed Execution. Regardless of the interface, the general principles behind job submission are the same.

A server machine is required to control your resources. The server will require the Campus Grid Toolkit Server software to be installed on it. This software can be downloaded from the OMII-UK website (www.omii.ac.uk/downloads).

The server machine is controlled by you through your client machine – generally your desktop computer – which will require an installation of the Campus Grid Toolkit Client software. This software is prepared and configured for you by your Campus Grid Toolkit Server, and is available for download from your server once the server software has been installed.

If your job requires the running of a program, that program must be installed on the resource on which you are running the job. The program may already be installed on the resource, or you may have to install the program yourself – either manually, or by including the installation of the program as part of your job.

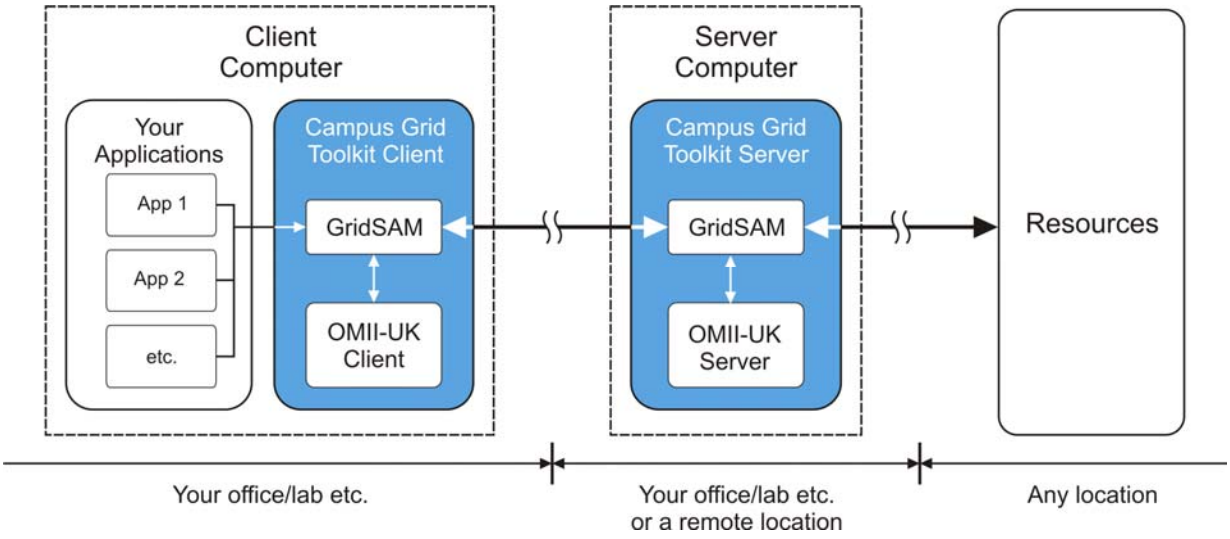
The Campus Grid Toolkit software is based on the OMII-UK Development Kit Server and Client infrastructure, so further information about the underlying structure of the software can be found in the OMII-UK Development Kit User guide (www.omii.ac.uk/dissemination).

4.1 The Direct GridSAM Interface

As described above, there are two interfaces to the Campus Grid Toolkit. The Direct GridSAM interface provides you with direct control over the GridSAM, which is a job submission Web Service that controls how your job is submitted to your resources. Directly interfacing with GridSAM provides you with close control over job submission. GridSAM only understands jobs submitted in one language: JSDL (job submission description language).

Once you have prepared your job on your client machine, your job will be sent to GridSAM on your server. GridSAM uses one of its in-built interfaces to process the job into a form recognised by the resource that you wish to use. The current version of the Campus Grid Toolkit supports three different resources: Condor, PBS and fork. As described in section 12, support for further types of resource will be added in future releases of the software. Once the job is complete, your results can be sent back to your client machine, or any other machine that you specify.

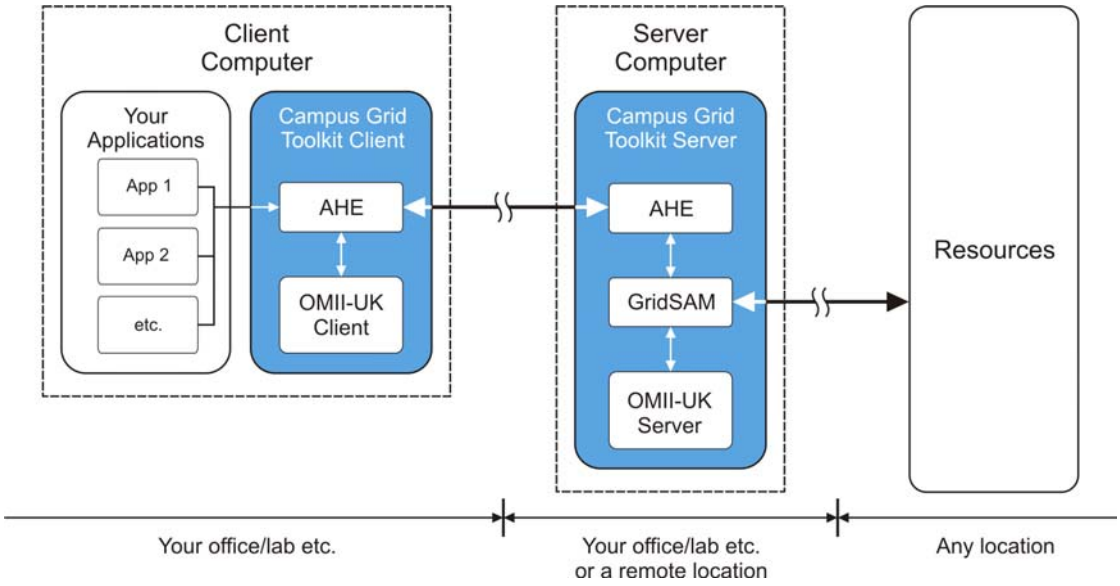
The diagram on the following page shows how the different components in the Campus Grid Toolkit interoperate when the Direct GridSAM interface is chosen.



4.2 Managed Execution Interface

Like the Direct GridSAM interface, the Managed Execution interface uses GridSAM to submit jobs to resources, but uses the Application Hosting Environment (AHE), rather than GridSAM, to interface with the user. AHE does not control job submission, it provides a generic set of commands for describing a job. It is relatively easy to call these generic commands from a different language, so AHE provides a route to making your job portable between different job submission Web Services (and using more job submission Web Services could potentially mean that you can access more resources).

The standard installation of the Campus Grid Toolkit with the Managed Execution interface will provide GridSAM for job submission and will configure AHE to talk to GridSAM. The following diagram shows the interaction of the Campus Grid Toolkit components when the Managed Execution interface is chosen.

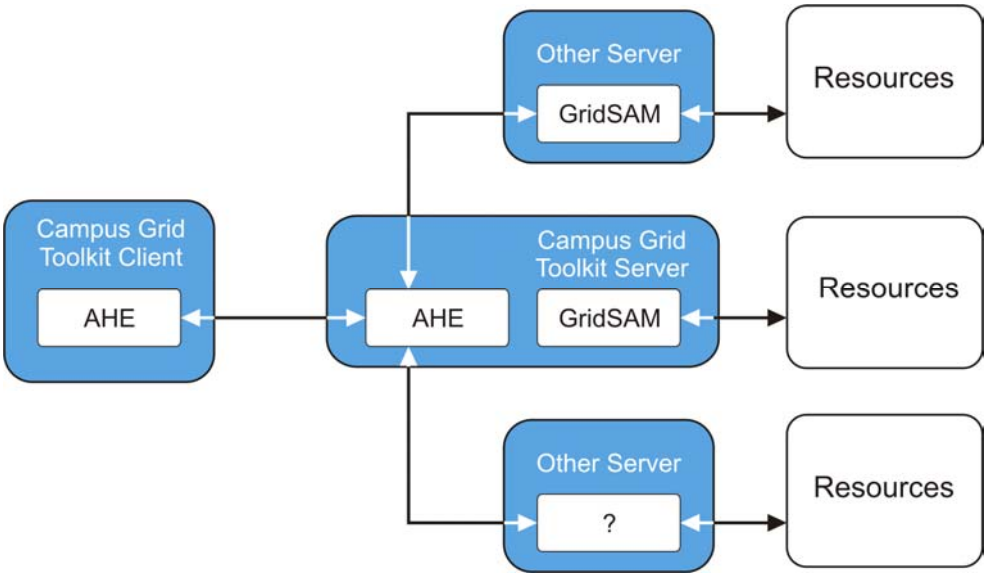


5 Which Interface to Choose?

In general, if you intend to use more than one type of resource, you should choose the Managed Execution interface, and if you intend to interface with only one type of resource, you should choose the Direct GridSAM method. Sections 5.1 and 5.2 provide a short overview of the benefits of each interface, which should help in your choice. Alternatively, you can install two Campus Grid Toolkits, one with each interface, so that you can choose which interface to use dependent on the type of job that you intend to submit.

The Direct GridSAM interface gives you complete control over GridSAM, meaning that you have the greatest level of control and flexibility for submitting your jobs to your chosen resource. However, the Direct GridSAM interface cannot be used to submit jobs to Web Services other than GridSAM.

The Managed Execution interface describes jobs using generic commands. The finite number of commands means that there can be an effect on the flexibility with which your jobs can be described. However, describing your job in these generic terms makes it easy to translate your job into different languages which will make it easier to access resources controlled by job submission Web Services other than GridSAM. This concept is summarised in the following diagram, in which the same job is submitted to a standard Campus Grid Toolkit Server running GridSAM , a non-OMII-UK Server running GridSAM and another server running a non-GridSAM Web Service.



5.1 Direct GridSAM Interface

- Best option if you have only one type of resource at your disposal
- Direct control of the GridSAM Web Service allows more flexibility and control over the submission and execution of your job
- Requires an understanding of GridSAM's language: job submission description language (JSDL)

5.2 Managed Execution Interface

- Easier to integrate into existing applications
- Best option if you intend to use different types of resource i.e. Condor, PBS, Globus, etc.
- Level of abstraction between yourself and GridSAM makes it easier to submit jobs that are portable between different types of resource, but may reduce the flexibility of job description
- AHE's protocol is more suited to users that do not have experience of the job submission description language
- Offers a separation between a simple client for use by researchers and a managed server as configured by an administrator.

6 Using Campus Grid Toolkit

As described in section 4, the purpose of Campus Grid Toolkit is to submit jobs to resources in your Campus Grid. Batch processing jobs are the most suitable type of job for the current version of Campus Grid Toolkit. Jobs that are most suitable for Campus Grid Toolkit are those that are most suitable for batch processing.

To submit a job, you must describe it in the appropriate language. This task will not be beyond the means of most researchers with some programming experience. Some knowledge of Perl (or a similar language) or Java will be required.

The example applications are the best starting point for developing a 'real' application of your own. The examples applications can be studied to understand the best way to submit jobs, or can be built upon with your own applications.

Details for obtaining further help with Campus Grid Toolkit can be found in section 13.

7 Quickstart

7.1 Notes

1. We recommend that you download the client installer file directly to the machine on which you intend to install the client software. If you don't do this, you will need to update the **registry.properties** file, as described in section 10.4.1.

2. Campus Grid Toolkit Client supports UNIX and Windows (Windows XP only).

3. Windows users must download and install Perl 5.8 for Windows from www.activestate.com.

4. The Campus Grid Toolkit Client cannot at present be installed on a path which includes spaces i.e., "`C:\Program Files`" or "`/home/user/my clients`" are unsuitable. This problem can be overcome for Windows users by using the short form notation, e.g. "`C:\progra~1`".

7.2 Quickstart Instructions

The following instructions describe how to perform a default installation of the Campus Grid Toolkit Server and Client. It is suggested that novice users read the full installation instructions (section 8) since they provide the process in more detail.

Note: a non-default installation can be performed by editing the specification (spec) files before installation of the Campus Grid Server. This process is described in section 11.

1. Install J2SE JDK version 5.0 or later.

2. Set environment variables:

```
[UNIX] export JAVA_HOME=<JAVA DIRECTORY>
```

i.e. provide the full path to the Java directory (e.g. `export JAVA_HOME=/usr/java/jdk1.5.0_13`).

```
[UNIX] export PATH=$JAVA_HOME/bin:$PATH
```

`[Windows]` Set the environment variables under the Advanced tab in the System Properties dialogue.

3. Download the installer for the Campus Grid Toolkit Server from the OMII-UK website (www.omii.ac.uk/downloads).

4. Untar the server installer, change to the **campus-grid-toolkit-server-1.1.3** directory and run the installer:

```
./CampusGridToolkitServerInstall.pl
```

5. Choose whether to install the Direct GridSAM or Managed Execution interface (see section 6 for details).

6. The following will be displayed when the installer has finished installing the server if you have chosen the Direct GridSAM interface:

```
** server installation and configuration complete
** server available at https://<YOUR MACHINE>:18443
```

If you have chosen the Managed Execution interface, the following will be displayed:

```
** server installation and configuration complete
** server available at http://<YOUR MACHINE>:18080
```

7. The pre-configured installer for the Campus Grid Toolkit Client is obtained from the URL identified at the end of the server installation. Go to the URL and click on the link to download the installer to the machine on which you wish to install the client software.

8. Unpack the client installer, change to the **campus-grid-toolkit-client-1.1.3** directory and run the installer (CLI and GUI versions are both available):

```
[UNIX] ./CampusGridToolkitClientInstall.sh
 or
[UNIX] ./CampusGridToolkitClientGuiInstall.sh

[Windows] CampusGridToolkitClientInstall.bat
 or
[Windows] CampusGridToolkitClientGuiInstall.bat
```

9. The installer will lead you through the rest of the installation.

8 Installation

The installation of the Campus Grid Toolkit has been designed to be as straightforward as possible. There are three mains steps: install Java, install the Campus Grid Toolkit Server on the server machine and then install the Campus Grid Toolkit Client on the client machine (typically, the client and the server are installed on different machines).

The installer for the Campus Grid Toolkit Server is found on the OMII-UK website (see below for details). Once installed, the server provides access to client software that has been specially pre-configured to interface with that server.

Note: Both the server and client should be installed as a normal user, i.e. not the root user.

Note: a non-default installation can be performed by editing the spec files before installation of the Campus Grid Toolkit Server. This process is described in section 11.

8.1 Downloading Java

Campus Grid Toolkit works with Java J2SE JDK version 6.0 - the latest version of Java, or Java J2SE JDK version 5.0 (with a minor caveat: there is a known issue with Sun Java 1.5.0_10 with the server).

Firstly, a quick explanation of Java. J2SE JDK version is the Standard Edition (SE) of the Java Development Kit (JDK). The version naming strategy was changed after Java version 1.4, which means that what would have been called ‘Java version 1.5’ is typically called ‘Java version 5.0’. Confusingly, both naming strategies are used on the Sun website!

To download Java, go to the Sun Developer Network Java website (www.java.sun.com) and select ‘Java SE’ from the downloads menu. Select the download for Java 6 (the most recent version) from the list that is presented. (Alternatively, if you wish to install Java 5.0, follow the link to ‘Previous Releases’ at the top of the list.)

8.2 Installing Java and setting JAVA_HOME and PATH

Instructions for installing Java can be found on the Sun Java website. After installation, it is necessary to tell your machine where to look when a Java command is invoked. The JAVA_HOME and PATH environment variables are used for this purpose.

8.2.1 Setting JAVA_HOME and PATH on a UNIX system

The recommended method for setting these variables is to enter the required commands into your **.bashrc** file. This option will work if you use the bash shell (most users do), If you do not use the bash shell, or if you do not know which shell you use, ask a system administrator for an alternative option. The **.bashrc** file runs each time that you log on to your machine, which means that the environment variables will be set automatically when you log on. (Alternatively, the commands can be entered directly into the terminal window, but this means that you must remember to set the commands each time that you open a new terminal.)

1. Install Java in line with Sun’s installation instructions.

2. Open a terminal window and open the **.bashrc** file for editing:

```
nano ~/.bashrc
```

Note: 'nano' is just one of the editors available under Linux, and may, or may not, be supplied with your distribution. If nano does not work, ask your system administrator for the details of an installed editor.

3. Set the JAVA_HOME environment variable by entering the following command onto a new line in your **.bashrc** file:

```
export JAVA_HOME=<JAVA DIRECTORY>
```

Where `<JAVA DIRECTORY>` is the full path to the directory in which Java was installed (e.g. `export JAVA_HOME=/usr/java/jdk1.5.0_13`).

4. Set the PATH environment variable by entering the following command onto a new line in your **.bashrc** file:

```
export PATH=$JAVA_HOME/bin:$PATH
```

5. Save and close the **.bashrc** file, then source it by entering the following command:

```
source ~/.bashrc
```

6. Enter the following command:

```
java –version
```

If your display shows your Java's version details, everything is working correctly.

8.2.2 Setting JAVA_HOME and PATH on a Windows system

1. Right click on My Computer in the Start menu.

2. Select 'Properties' from the resulting menu to open the System Properties dialogue and click on the 'Advanced' tab.

4. Click on the 'Environment Variables' button.

5. Add `JAVA_HOME` as either a User or System variable by clicking on the appropriate 'New' button and entering `JAVA_HOME` as the 'Variable Name' and the full path to the directory in which Java was installed (e.g. `export JAVA_HOME=/usr/java/jdk1.5.0_13`) as the 'Variable Value'. Click 'OK'.

6. Within the list of System variable, locate the `Path` variable, double-click it to open it for editing and add `%JAVA_HOME%\bin;` to the beginning of the `Path` variable's 'Variable Name'. Click OK.

7. Click 'OK' to close the System Properties dialogue.

8.3 Security Certificates

The Campus Grid Toolkit Server and Client both use an X.509 certificate for security purposes. During installation of the software, a temporary (30 day) certificate is issued for use whilst setting up and familiarising yourself with the software. After the temporary certificate expires, it is necessary to obtain a longer term certificate from a Certification Authority, such as the National Grid Service. Further details of this process can be found in the OMII-UK Development Kit Installation guide (www.omii.ac.uk/dissemination).

8.4 Installing the Campus Grid Toolkit Server

1. Download the Campus Grid Server installer (**campus-grid-toolkit-server-1.1.3.tar**) from the OMII-UK website (www.omii.ac.uk/downloads).

2. Untar the installation files:

```
tar xf campus-grid-toolkit-server-1.1.3.tar
```

This will create a new directory, called **campus-grid-toolkit-server-1.1.3**, within the current directory.

3. Change to the **campus-grid-toolkit-server-1.1.3** directory and run the server installer:

```
cd campus-grid-toolkit-server-1.1.3

./CampusGridToolkitServerInstall.pl
```

4. You will be asked to specify a directory in which the server should be installed. For example:

```
/home/<USER>/campus-grid-toolkit-server
```

Where <USER> is your username or the name of the account that you are using.

5. You will be asked whether you wish to install the Direct GridSAM or Managed Execution interface. For managed Execution you can choose from two different database systems. Hypersonic is a pure Java database system, which is recommended for small or test systems. Postgres is a well-known database system, and is recommended for larger or production systems.

For help on choosing an interface, see sections 4 and 5 of this guide.

You will be asked which connector is to be used by GridSAM. The connector will be fork for local execution of jobs, Condor, for submitting jobs to your local Condor pool or PBS, for submitting jobs through your PBS system.

If Condor is chosen, the installer will attempt to locate your Condor installation by looking at the environment for a CONDOR_HOME value, which if defined will point to directory, e.g. /home/condor/bin. In the case of the ICT Condor connector, the condor binaries must be on the PATH.

In the case of PBS and ICT PBS, the environment variables PBS_HOME, PBS_QUEUE and PBS_SPOOL_DIR are honoured. If you know the values of any of these variables, then it is best to set them in your environment prior to running the installer otherwise default values will be chosen.

If you need to change the GridSAM connector settings at any point post-installation, then you can run the supplied script <SERVER>/bin/manageGridsamConnectors.pl. See section 8.5 for more details on how to use this script.

6. If you have chosen the Direct GridSAM interface, the following will be displayed when the installer finishes:

```
** server installation and configuration complete
** server available at https://<YOUR MACHINE>:18443
```

If you have chosen the Managed Execution interface, the following will be displayed:

```
** server installation and configuration complete
** server available at http://<YOUR MACHINE>:18080
```

Where <YOUR MACHINE> is the full name of your machine (i.e. it will contain the machine name and the domain name).

8.5 Configuring and reconfiguring the Campus Grid Toolkit Server DRM Connectors

To help you configure your server to use different back-end resources, such as Condor or PBS there is a script to help you do this more easily.

This script is called manageGridsamConnectors.pl and is located in the server's bin directory. This script is used as part of the installation process, but is also used for changing the settings on an existing server. Bofore using this script, change to the bin directory.

```
cd <SERVER>/bin
```

You can get simple usage information from the script by running it with no command line arguments.

```
$ ./manageGridsamConnectors.pl
./manageGridsamConnectors.pl <fork|condor|pbs|ict-condor|ict-
pbs> <nobounce>?

The above command sets the connector to be used by GridSAM and restarts the
container with the new settings applied. Use nobounce to suppress this.

fork     - The fork connector sets up execution to be performed locally by GridSAM.
condor   - The condor connector will use the CONDOR_HOME environment
           variable to locate the Condor binaries.
pbs      - The PBS connector will use the PBS_HOME, PBS_QUEUE and
           PBS_SPOOL_DIR environment variables to locate the PBS binaries and
           to set the name of the queue and spool directory to use.
ict-condor - The ICT Condor connector will use the CONDOR_SPOOL_DIR
           environment variable to set the name of the spool directory to use,
           this will default to /tmp if not set.  Please note that the condor
           binaries, such as condor_submit should be on your PATH.
ict-pbs  - The ICT PBS (GOS) connector will use the PBS_QUEUE and
           PBS_SPOOL_DIR environment variables to set the name of the queue
           and spool directory to use. Please note that the spool directory
           should be on shared storage between the nodes.
```

In order to get the connectors configured, in most cases you also need to prime the environment with a few details before running the script.

Fork Connector - This is for executing jobs locally and does not require any environment change.

Condor Connector - This needs CONDOR_HOME set to be the directory where the Condor binaries are stored, this is likely to be something similar to /home/condor or /usr/local.

PBS Connector - This is the original PBS connector. This require PBS_HOME, which is typically /usr/local/bin. PBS_QUEUE_NAME, which is usually 'batch' by default. And finally PBS_SPOOL_DIR, which should be a fully qualified path to your shared storage, as used by your PBS system.

ICT Condor Connector - This is an alternative connector supplied by ICT, which is a new addition in CGT 1.1.3. It requires that the condor binaries such as condor_q be on the PATH.

ICT PBS Connector - This is an alternative connector supplied by ICT, which is a new addition in CGT 1.1.3. It requires PBS_QUEUE and PBS_SPOOL_DIR to be set in the same way as the PBS Connector. This connector will use PATH in place of PBS_HOME to find the PBS binaries such as qstat.

The optional second argument to manageGridsamConnectors.pl is 'nobounce'. Usually when the script needs to be run, the server is already running, so its normal operation is to shut down the server, alter the configuration and restart it. If the 'nobounce' argument it set, then only the configuration is changed. This makes sense if you are making a set of changes and the server is already shut down.

8.6 Installing the Campus Grid Toolkit Client

Unlike the Server software, the Campus Grid Toolkit Client is available for both UNIX (i.e. Linux and Mac) and Windows operating systems. The UNIX and Windows clients are both available for download from the server, once it has been installed. There are two version of client installer available: a CLI (command line interface) version and a GUI (graphical user interface) version.

8.6.1 Notes

1. We recommend that you download the client installer file directly to the machine on which you intend to install the client software. If you don't do this, you will need to update the **registry.properties** file, as described in section 10.4.

2. Campus Grid Toolkit Client supports UNIX and Windows (Windows XP only).

3. Windows users must download and install Perl 5.8 for Windows from www.activestate.com.

4. The Campus Grid Toolkit Client cannot at present be installed on a path which includes spaces i.e., "`C:\Program Files`" or "`/home/user/my clients`" are unsuitable. This problem can be overcome for Windows users by using the short form notation, e.g. "`C:\progra~1`".

8.6.2 Installation Instructions

The installer for the Campus Grid Toolkit Client is obtained from the Campus Grid Toolkit Server, which is located by the URL given at the end of the server installation process. Select the UNIX or Windows client as appropriate.

1. Go to the URL given at the end of the server installation. (You may be asked whether you want to accept the server's security certificate. You must accept the certificate to proceed with the installation.)

2. A web page will be displayed. Click on the appropriate link to download the Campus Grid Client software for UNIX (**campus-grid-client-toolkit-1.1.3.tar**) or Windows (**campus-grid-client-toolkit-1.1.3.zip**) to your home directory.

3. `[UNIX]` Unpack and run the installer for the Campus Grid Toolkit Client:

```
[UNIX] cd ~

[UNIX] tar xf campus-grid-toolkit-client-1.1.3.tar

[UNIX] cd campus-grid-toolkit-client-1.1.3

[UNIX] ./CampusGridToolkitClientInstall.sh
```

or

```
[UNIX] ./CampusGridToolkitClientGuiInstall.sh
```

4. `[Windows]` Unzip and run the installer for the Campus Grid Toolkit Client:

`[Windows]` Find the **campus-grid-client-toolkit-1.1.3.zip** file and extract the files (using File Explorer, right click the file, select 'Extract all' from the resulting menu and then enter your chosen path in the extract wizard).

`[Windows]` Using File Explorer, open the campus-grid-toolkit-client-1.1.3 directory and double-click the **CampusGridToolkitClientInstall.bat** file or **CampusGridToolkitClientGuiInstall.bat** file.

5. The installer will ask for an installation path. If you press return at this point, a recommended path will be entered. Alternatively, if you would prefer a different path, enter the details and press return.

6. The installer will ask for http and then https proxy details. These details can be obtained from your system administrator. If you do not use a proxy, simply press return.

7. The installer will ask for your machine name. This is the fully qualified name (i.e. it contains a domain name) given to your machine so that it can be identified on the network. The name suggested by the installer is typically correct, with one notable exception: if the installer thinks that your machine's name is 'localhost', you may need to ask your system administrator for help.

8. The installer asks for details to create a keystore (your organisation, email etc.). The keystore provides information that is used to securely identify the client to the remote server.

9. The installer will lead you through the rest of the installation.

8.7 Starting and Stopping the server

If you reboot your machine, you will have to re-start the server and database. Assuming that you installed the server at `/home/<USER>/campus-grid-toolkit-server`, start the server by entering the command:

```
/home/<USER>/campus-grid-toolkit-server/bin /startomii.sh

(cd /home/<USER>/campus-grid-toolkit-server/bin && ./hypersonic_control.pl
start)
```

For Hypersonic, or for Postgres use:

```
/home/<USER>/campus-grid-toolkit-server/database_control start \
      /home/<USER>/campus-grid-toolkit-server/postgres
```

The server can be stopped by entering the command:

```
/home/<USER>/campus-grid-toolkit-server/bin/stopomii.sh

(cd /home/<USER>/campus-grid-toolkit-server/bin && ./hypersonic_control.pl
stop)
```

For Hypersonic, or for Postgres use:

```
/home/<USER>/campus-grid-toolkit-server/database_control stop \
      /home/<USER>/campus-grid-toolkit-server/postgres
```

Note: ensure that JAVA_HOME is set and that Java is on your path, see section 8.2 for details.

9 Example Applications

The example applications can be used to check the installation and operation of the Campus Grid Toolkit, and to demonstrate what Campus Grid Toolkit can do. They can also be used as the starting point for developing your own 'real' applications.

Four example applications are provided with the Direct GridSAM version of Campus Grid Toolkit and two example applications are provided with the Managed Execution version. The example applications for Direct GridSAM (section 9.1) will only work if your Campus Grid Toolkit Server is set for Direct GridSAM (see step 5 in section 8.3 above). The same rule applies for the Managed Execution example applications.

9.1 Direct GridSAM Example Applications

Note: When the example applications run, the remote server will open a connection to the client. Therefore it may be necessary to open a port on the client machine's firewall. The following example applications use port 3333.

9.1.1 Uname (UNIX only)

The Uname example application submits four jobs to the Campus Grid. Each job (`/bin/uname -a`) requests and then displays the name of the node on which it ran. This simple demonstration allows you to check the operation of Campus Grid Toolkit without the extra complication of requiring a graphical interface or significant resources.

1. Change to the directory that's identified at the end of the client installation (typically, **~/campus-grid-toolkit-client/gridsam**).

2. Change to the `uname` directory and run the example application.

```
cd uname

./uname.pl
```

The terminal window will display information that will show that the jobs have been submitted and, after a short while, that the jobs have completed. The names and some other details of each of the machines in your Campus Grid will then be displayed.

If you have not connected the Campus Grid Toolkit to any machines in your Campus Grid, all jobs will be executed on the same machine as the server, i.e. local execution. Consequently, the only machine name that will be returned is the name of your server's machine.

As an alternative to `uname.pl`, there is also `uname_webdav.pl`.
This variant of the script uses the WebDAV service installed as part of Tomcat on the CGT server. The inputs and outputs from the job are staged via WebDAV instead of using FTP.

```
./uname_webdav.pl
```

9.1.2 Uname Java (UNIX and Windows)

Uname Java is a purely Java example of the above Uname example application. It operates in the same manner, but uses the GridSAM Java API directly.

1. Change to the directory that's identified at the end of the client installation (typically, **~/campus-grid-toolkit-client/gridsam**).

2. Change to the `uname` Java directory and run the example application.

```
cd uname_java
```

```
./RunGridsamUnameJavaClient.sh
```

or, on Windows:

```
./RunGridsamUnameJavaClient.bat
```

9.1.3 Fractal (UNIX only)

Fractal calculates the visualisation of a Java-based Mandelbrot set simulation. The Mandelbrot set calculation is divided into a number of jobs, which are sent to your Campus Grid for processing. Each node that processes a job produces an image containing part of the Mandelbrot set which is over-stamped with the node's machine name. The final image is assembled by tiling the resulting images into a complete rendition of the Mandelbrot set.

1. Change to the directory that's identified at the end of the client installation (typically, **~/campus-grid-toolkit-client/gridsam**).

2. Change to the fractal directory and run the example application

```
cd fractal
```

```
./fractal.pl
```

3. The job's progress will be shown in the terminal window and a new window, called 'Tile Image Client', will open.

4. The final image will show a Mandelbrot set overlaid with the names of the machines that calculated each tile in the image.

5. The fractal example application can accept different parameters (1 to 5, i.e. `./fractal.pl 3`) that will change the appearance of the Mandelbrot set.

In addition to `fractal.pl` there is also `fractal_webdav.pl`. This file operates in the same way as `fractal.pl` except the file staging is performed via WebDAV rather than FTP.

```
./fractal_webdav.pl
```

9.1.4 POV-Ray (UNIX only)

POV (Persistence of Vision) is a ray-tracing package. The POV-Ray example application is based on the 3D rendering of 24 frames of a moving image. The rendering for each frame of the image is divided between the nodes in your Campus Grid. The rendered images may be viewed as a movie (e.g. using animate under Linux).

Note: POV-Ray only works on Intel-based Linux servers.

1. Change to the directory that's identified at the end of the client installation (typically, **~/campus-grid-toolkit-client/gridsam**).

2. Change to the pov directory and run the example application

```
cd pov
```

```
./pov.pl
```

3. The progress of the job will be displayed in the terminal window. When the job is finished, all 24 frames of the moving image will be saved in the `pov/ftpdata` directory as .png files.

9.2 Managed Execution Example Applications

9.2.1 Installing the "Managed Execution" (AHE) example server-side binaries.

Applications that run using AHE must be pre-installed. Typically, 'real-life' applications will be large and complex, so ad hoc installation is not a sensible option. The example applications follow this rule and, as such, they must be installed before use.

The following sections of this guide describe how to install the AHE Example Applications (or how to direct AHE to a previously installed version of the application) and how to run the example application.

9.2.2 Installing the BuddhaBrot Example Application

The BuddhaBrot example application must be installed in a standard location on each execution node in your cluster, if your are using Condor, or on your server, if you are not using Condor. The BuddhaBrot example application comes with the Campus Grid Toolkit Server installer. The source distribution file **buddhabrot_source.tgz** file is located within the `campus-grid-toolkit-server-1.1` directory. This is a source distribution, however, for convenience, it also contains a pre-built 32 bit Linux binary which is suitable for most Linux platforms.

1. Unpack the **buddhabrot_source.tgz** file:

```
tar xvzf buddhabrot_source.tgz
```

2. To use the pre-built binary, copy the file **buddha** to the `/usr/local/bin/` directory on your server, or to each of the execution nodes in your Condor pool.

If `/usr/local/bin` is not a convenient location, an alternative can be used. However, you must direct AHE to this location be editing a wrapper script (**buddha.sh**) located in the top level of your server installation.

1. Open the **buddha.sh** file for editing

2. Locate the following line:

```
BUDDHA=/usr/local/bin/buddha
```

3. Edit this line to match the location of your BuddhaBrot executable.

4. Save and close the **buddha.sh** file.

Instructions for building your own buddha binary can be found in the README file that is located within the unpacked sources.

9.2.3 Running the BuddhaBrot Example Application (UNIX and Windows)

BuddhaBrot is a similar application to the Fractal application described above. BuddhaBrot calculates a number of different fractals at different angles relative to each other, and then superimposes the fractals together to produce an image. The resulting image is reputed to look like Buddha. The calculation for each fractals is described as a different job and calculated by a different node in your Campus Grid.

1. Change to the directory that's identified at the end of the client installation (typically, **~/campus-grid-toolkit-client/ahe**).

2. Change to the buddhabrot directory and run the example application

```
cd buddhabrot
```

```
./buddhabrot_client.pl
```

3. The job's progress will be shown in the terminal window.

4. A new terminal window will open to show the final image of the BuddhaBrot.

9.2.4 Installing the POV-Ray Example Application

If you are using Condor or PBS, POV-Ray must be installed in a standard location on each execution node in your cluster. The default location for installing POV-Ray is `/usr/local/bin/povray`. Alternatively, if you already have POV-Ray installed, you can either link the binary into this location or edit the server configuration to match its current location. This process is described in section 9.2.5.

1. Ensure the gcc C compiler is available on your system by typing:

```
gcc –v
```

If your display shows your gcc C compiler's version details, everything is working correctly. If the compiler is not installed, it can be obtained from the install media for your operating system.

2. Download the POV-Ray 3.61 source code archive (**povray-3.6.tar.bz2**) from www.povray.org/download/ and unpack it using the command:

```
tar xvjf povray-3.6.tar.bz2
```

3. Change to the unpacked directory and enter the following configure command, adding your name and email address as appropriate:

```
cd povray-3.6.1
```

```
./configure COMPILED_BY="A. N. Other <another@example.com>" --
prefix=/usr/local
```

4. Build and install POV-Ray by entering the following commands:

```
make
```

```
su -
```

```
cd /<PATH TO POVRAY SOURCE>/povray-3.6.1
```

```
make install
```

Where <PATH TO POVRAY SOURCE> is the path to the **povray-3.6.1** directory

5. Test the POV-Ray installation by entering the following commands:

```
/usr/local/bin/povray +Otest.png +W800 +H600 +A0.3 \

/usr/local/share/povray-3.6/scenes/advanced/gaussianblob.pov

display test.png
```

If you do not have the 'display' application, use any application that can open png files.

9.2.5 Editing a previously installed POV-Ray

If you already have POV-Ray installed, you can either link the binary into this location or edit the server configuration to match its current location. A wrapper script (called **pov.sh**), which is uploaded to every execution node prior to job execution, is used for this purpose. The **pov.sh** file can be found in the top level directory of your server installation (typically `campus-grid-toolkit-server`).

1. Open the **pov.sh** file for editing

2. Locate the following line (near the top of the file):

```
POVRAY=/usr/local/bin/povray
```

3. Edit this line to match the location of your povray executable.

4. Save and close the **pov.sh** file.

9.2.6 Running the POV-Ray Example Application (UNIX and Windows)

The Managed Execution version of POV-Ray is similar to that described in section 9.1.4.

1. Change to the directory that's identified at the end of the client installation (typically, **~/campus-grid-toolkit-client/ahe**).

2. Change to the povray directory and run the example application

```
cd povray

./pov_client.pl
```

3. The progress of the job will be displayed in the terminal window. When the job is finished, an animation will be shown that comprises the number frames specified when the application was invoked.

10 Configuring Campus Grid Toolkit

10.1 Introduction

As described at the start of this guide, the installation of Campus Grid Toolkit is intended to be straightforward. This ease-of-installation is made possible by automatically setting the configurable aspects of the software to popular default values. The default values used by the installer are stored in a file called **setup.spec** (.spec is a contraction of 'specification', so that these files are known as 'spec files').

Advanced users may wish to change these defaults. It is the purpose of this section to describe how such changes can be made. Section 11.3 provides an annotated description of the **setup.spec** to help users understand the configurable aspects of Campus Grid Toolkit. The commands available in the scripting language are described in Section 11.5.

The current version of Campus Grid Toolkit makes use of either OMII-UK's GridSAM software component or OMII-UK's GridSAM and AHE software components. Experienced OMII-UK users may wish to add further OMII-UK software components. The process for adding further software components is described in section 11.4.

10.2 The setup.spec file with annotations

The **setup.spec** file is used as the main script for the installation of a server, and is found in the **campus-grid-toolkit-server-1.1.3** directory. A slightly simplified version with some annotations is included below.

The first section in the spec file determines which ports will be used by Tomcat (the Web Services container). The values for these ports can be changed if required. WebDav and Database are used by Managed Execution Server, but not by Direct GridSAM.

```
# which ports Tomcat should use
set OMII_TOMCAT_PORT=18443
set OMII_TOMCAT_SHUTDOWN_PORT=18005
set OMII_TOMCAT_AJP_PORT=18009
set OMII_WEBDAV_PORT=18080
set OMII_DATABASE_PORT=15432
```

The next section deals with the OMII_HOME variable, which reflects the location for installing the server. By default the user will be prompted to enter a directory unless OMII_HOME has been set in the calling environment. However, a predefined location can be used instead, by commenting out the 'read' line, uncommenting the 'set' line and then using the 'set' line to provide a predefined location.

```
# install to a set location, or ask the user to specify one
# set OMII_HOME=$HOME/campus-grid-toolkit-server
read -if-not-set OMII_HOME Please specify a directory where you would like the
server installed:
```

The OMII_HOSTNAME variable is the name by which your server machine will be referred to. The installer tries to establish this value automatically, else the user is asked

```
set OMII_HOSTNAME=$HOSTNAME
set -if-not-set -exec OMII_HOSTNAME=hostname
read -if-not-set OMII_HOSTNAME Please enter your machine's the fully qualified
hostname:
```

The next section of the spec file deals with the generation of the server's X.509 temporary security certificate. The default values can be changed to reflect your own details.

```
# generate the OMII temporary certificate - please change to your own details
set KEY_O=OMII-UK
set KEY_OU=Development
set KEY_L=Southampton
set KEY_S=Hampshire
set KEY_C=UK
set KEY_email=$USER@$OMII_HOSTNAME
exec extension/get_details.pl quiet CN=$OMII_HOSTNAME O=$KEY_O OU=$KEY_OU
L=$KEY_L S=$KEY_S C=$KEY_C emailAddress=$KEY_email outfile=extension/key.url
```

The following section provides the user with the choice between Managed Execution and Direct GridSAM.

```
# install the server
echo Would you like your server to use Direct GridSAM or Managed Execution?
echo   Direct GridSAM will install GridSAM and associated components,
echo   whereas Managed Execution will install and enable AHE to be used as a
front end to GridSAM.
echo 1) Direct GridSAM (default)
echo 2) Managed Execution with Postgres
echo 3) Managed Execution with Hypersonic

read MODE
map RESULT MODE 1:basic,2:ahe-postgres,3:ahe-hypersonic basic
runtime_include install_campus_grid-$RESULT.spec
```

10.3 Installing further software components

Experienced users of OMII-UK's software may wish to install further software components. This can be achieved by adding further install commands to the **install_campus_grid-basic.spec** file or the

install_campus_grid-ahe-common.spec, both of which are found in the **campus-grid-toolkit-server-1.1.3** directory.

10.4 Re-configuring the client

Typically, the only changes that are made to the Campus Grid Toolkit Client's configuration are the addresses of the client and server machines. This information is held in two files – one is used by the Direct GridSAM server and the other is used by the Managed Execution server. The files and instructions for making changes are discussed below.

10.4.1 Configuring the Direct GridSAM Client

Configuration of the Direct GridSAM Client is performed by editing the **registry.properties** file found in the top level of your client installation. The file has the following contents:

```
remote_server_url=https://<YOURSERVER>:18443
remote_server_webdav_url=http://<YOURSERVER>:18080
default_client_ip_address=<IPADDRESS>
FtpServer.server.config.self.host=<IPADDRESS>
```

Where <YOURSERVER> is the fully-qualified address of the server on which your Campus Grid Toolkit Server is installed, and <IPADDRESS> is the public-facing IP address of your client machine (as seen by your Campus Grid Toolkit Server).

If you download the client installer file on a first machine, then install it on a different machine, the <IPADDRESS> will be incorrect. This is because <IPADDRESS> is automatically set by your Server to the address of the machine that downloads the Client. In this case, <IPADDRESS> will have to be changed so that it represents the machine on which you installed the Client.

10.4.2 Configuring the Managed Execution Client

If you wish to change the server used by your client, you will need to edit the **aheclient.properties** file, which is found in the ahe/config directory of your client installation. The server URL occurs three times in the **aheclient.properties** file, all three incidences must be updated.

10.5 Clearing out the GridSAM database

From time to time it is desirable to clear out the database held by GridSAM. The database stores the states of the jobs. Often, while a system is being commissioned, several test jobs are run, which means that a partially configured system can end up with a collection of jobs in inconsistent states. Installed in the `bin` directory is `clearGridsamDatabase.pl`, this script takes two parameters. Firstly, the directory of the installation you wish to clear and, secondly, a flag to indicate whether the container is to be shutdown while the clean-up takes place. The container should not be running when the database is cleared. However, if the script is to be run as part of some larger process, the container may already be shut down.

```
./clearGridsamDatabase.pl <omii-home> <nobounce>?
```

e.g.

```
./clearGridsamDatabase.pl /home/omii/campus-grid-toolkit-server nobounce
```

11 Future additions to functionality

Version 1.1.3 of Campus Grid Toolkit is the third incarnation of the software. In the future, we plan releases of new versions of the software that will provide further functionality. The following is a list of some of the additions that are currently being planned.

1. Support for more computational and data resources.

2. Portal support from GridSphere.

3. Further support for job monitoring.

12 Support and Further information

Campus Grid Toolkit is a fully supported software package. Further details on the operation of the OMII-UK Development Kit Server, GridSAM and AHE – the software that comprises Campus Grid Toolkit – can be found in the OMII-UK Development Kit User Guide (3.4.0), available from www.omii.ac.uk/disseminsation.

Support for Campus Grid Toolkit is provided on the OMII-UK website (www.omii.ac.uk/support) and from the support email address (support@omii.ac.uk).

13 Appendix: Available Scripting Commands

The following section contains a language reference for the spec files used during installation.

13.1 echo

Echo a message to the console. The message is evaluated so that variable substitutions can be performed.

```
echo <message>
```

Example:

```
echo “hello $USER”
hello bob
```

13.2 exec

Executes a command, writing any output to the console.

```
exec <command>
```

Exec commands may also redirect to files and any reads on standard input will be read from the console.

Examples:

```
exec uname
Linux

exec uname > tmp
exec cat tmp
exec rm tmp
>Linux
```

13.3 set

Set a variable to a specified value. The value is evaluated before it is assigned, so variable expansion can be performed.

```
set <-if-not-set>? <-exec>? <variable>=<value>
```

If the -if-not-set flag is set, then the assignment will only occur if the variable is not already defined.
If the -exec flag is set, the value is taken to be a shell command that should be executed, and its output will be the value assigned. Both flags may be used together.

Example:

```
set MESSAGE="hello $USER"
echo $MESSAGE
```

```
hello bob
```

13.4 read

Read a value from the console and assigns it to the variable. An optional message may also be supplied.

```
read <-if-not-set>? <variable>
```

and

```
read <-if-not-set>? <variable> <message>
```

If the -if-not-set flag is set, then the assignment will only occur if the variable is not already defined.

Example:

```
read NAME
echo "hello $NAME"
> bob
hello bob
```

or

```
read NAME "Please enter your name:"
echo "hello $NAME"
"Please enter your name:" bob
hello bob
```

13.5 install

Installs a server component, this can be a container, database or Commissioned Software component.

```
install <class> <bundle name>
```

Example, install GridSAM:

```
install component gridsam
```

Example, install the basic server:

```
install core ws-container
```

The `<class>` field can be core, database or component. Valid `<bundle name>` fields for the core and database are described in the following table.

class	bundle name	description
core	ws-container	Installs the basic server (Tomcat, Axis et. al.)
database	postgres	Installs a Postgres DBMS
database	postgres-cs	Sets up the Commissioned Software database user and tablespace.
database	hypersonic	Sets up the Hypersonic database system for use with AHE.

The name of the OMII-UK Software Component must match the name specified on the OMII-UK download repository and should be both lowercase and stripped of punctuation i.e. 'GridSAM' goes to '`gridsam`', 'WSRF::Lite' goes to '`wsrflite`'. See the group names column on www.omii.ac.uk/installation/components for more details.

13.6 include

Includes another spec file. This is useful for breaking up the parts of your server installation into logical parts.

```
include <spec file>
```

Example:

file A.spec
```
      echo “from file A”
      include B.spec
```

file B.spec
```
      echo “from file B”
```

```
from file A
from file B
```

13.7 runtime include

The runtime_include command is similar to include, except error checking is left until the execution phase, so variable substitution can be performed.

```
runtime_include <spec file>
```

Example:

```
runtime_include $FILE
```

Unlike standard include, the existence of the file referred to by $FILE cannot be verified before trying to execute its contents.

13.8 runtime exec

The runtime_exec command is used to take the value of a variable and execute it in a shell context.

Example:

```
read CMD “enter a command”
runtime_include CMD
```

13.9 exit

The exit command simply exits the installer, with an optional message.

```
exit <message>?
```

Example:

exit “Exiting the installer”

13.10 map

Supplies a way to conditionally set a variable based on the current value of a second variable. This, when used in conjunction with runtime_include, can be used to dynamically choose from a selection of additional spec files, then to execute the selected file’s contents.

```
map <result variable> <source variable> <key-value list> <default value>
```

Example:

```
echo Please select a size: 1) Small 2) Medium 3) Large
```

```
read INPUT
map RESULT INPUT 1:Small,2:Medium,3:Large Small
echo You selected $RESULT
runtime_include $RESULT.spec
```

The user will be presented with a choice, if, for example, '2' was entered:

```
Please select a size: 1) Small 2) Medium 3) Large
> 2
You selected Medium
```

Then Medium.spec would be loaded and executed.
If an answer that does not match 1, 2 or 3 (in this case) is given, then the default value is assumed. In this case, that would be 'Small'.

13.11 check port

Supplies a way to check the validity and availability of a port. So that that for instance the value contained within OMII_TOMCAT_PORT can be checked to see if it is a valid port reference (eg, a number in the valid range, rather than a string) and that the port is available for use.
If the port is invalid or not available then the installer will exit with an error message reflecting the nature of the problem encountered.

```
check_port <port>
```

Example:

```
check_port 18443
check_port $OMII_TOMCAT_PORT
```